Haute Ecole de Gestion & Tourisme
Hochschule für Wirtschaft & Tourismus

Filière Informatique de gestion
Studiengang Wirtschaftsinformatik

Travail de Bachelor 2017

Réalité augmentée pour les évènements sportifs en direct

Source : Adventures-Lab

Etudiant : Romain Coupy

Professeur : Antoine Widmer

Déposé le : 2 août 2017

HES-SO Valais-Wallis • rue de la Plaine 2 • 3960 Sierre
+41 27 606 89 11 • info@hevs.ch • www.hevs.ch

RÉSUMÉ

Ce document traite de la réalité augmentée dans le domaine des courses cyclistes. Nous nous intéresserons plus particulièrement du point de vue du spectateur qui aura l'opportunité d'avoir une perspective différente de la visualisation d'une course, autre que la télévision ou encore d'une application mobile représentant la position des athlètes sur une carte en deux dimensions.

Nous aborderons tout d'abord les concepts de réalité virtuelle et de réalité augmentée afin de bien comprendre les différences entre ces deux technologies.

Ensuite nous analyserons l'utilisation de la réalité augmentée dans le milieu sportif et leurs premières apparitions pour le public. Nous verrons à la suite de ce chapitre la même utilisation mais dans les courses cyclistes.

Une fois ces analyses effectuées, nous nous attarderons sur un état de l'art des moyens de réalité augmentée qui existent sur le marché à l'heure actuelle. Dès lors que ces outils sont définis, nous nous pencherons sur les logiciels de développement qui correspondent à nos critères, nous désirons créer une application s'appuyant sur une technologie de réalité augmentée qui touche les systèmes d'exploitation suivant : Android, iOS et Windows.

L'objectif principal de ce travail est de réaliser un prototype d'application mobile qui permettra à l'utilisateur de visionner une course cycliste depuis son smartphone en utilisant la réalité augmentée. L'application devra être à même de pouvoir récupérer des données fournis par une entreprise externe et ensuite, de transposer les coordonnées réelles des participants à la course sur une carte en trois dimensions.

La partie technique de ce projet a été réalisée de manière itérative basée sur la méthodologie Scrum.

L'objectif final sera de proposer à la société Adventures-Lab un prototype fonctionnel qui implémentera quelques fonctionnalités permettant de se rendre compte de la plus-value d'une telle application pour le spectateur.

Mots-clés : réalité virtuel et réalité augmentée, outils de réalité augmentée, sports et réalité augmentée, Adventures-Lab, système d'exploitation mobile, méthodologie Agile, Scrum.

AVANT-PROPOS

Ce travail a été proposé par la société Adventures-Lab, entreprise active dans le domaine de la réalité augmentée basée au Technopôle à Sierre, en collaboration avec la société GPST, GEO POSITIONING SWISS TECHNOLOGY basée dans le canton de Vaud. Ils ont émis une idée d'application utilisant la technologie de réalité augmentée afin d'ouvrir une nouvelle perspective de visualisation des courses cyclistes pour les spectateurs.

Le but de ce travail est de fournir à Adventures-Lab un prototype d'une application mobile capable d'afficher la position exacte des participants d'une course cycliste sur une carte en trois dimensions.

Pour ce faire, il a été établi un état de l'art des outils de réalité augmentée présent sur le marché ainsi que des logiciels de développement et moteurs de jeux. L'entreprise GPST fournit les données des athlètes et notre travail est de les récupérer pour ensuite les utiliser au sein de l'application mobile. La principale difficulté a été de trouver un moyen de transposer les coordonnées GPS réelles des coureurs sur notre modèle de carte en 3D.

Ce rapport fournit tout ce qui a été analysé ainsi que les étapes importantes du développement.

REMERCIEMENTS

Je tiens à remercier toutes les personnes m'ayant soutenu durant ce travail de Bachelor, particulièrement :

M. Antoine Widmer, professeur à la HES-So Valais/Wallis, pour m'avoir encadré et donné de précieux conseils durant le projet.

M. Thomas Crausaz, PDG d'Adventures-Lab, pour ses conseils dans la création du modèle de la carte en 3D.

M. Pierre-Albert Cantin, Directeur Général de GPST, ainsi que M. Peter Meier pour m'avoir fourni les données des athlètes.

M. Calixte Mayoraz, pour son aide dans une des parties techniques du code.

Mme. Nathalie Nanchen, Mme. Mélissa Bagnoud, M. Régis Bagnoud, pour leur participation à la relecture du rapport.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

GLOSSAIRE

- AR : Augmented Reality (réalité augmentée)
- VR : Virtual Reality (réalité virtuelle)
- UWP : Universal Windows Platform (Microsoft, s.d.)
- GPS : « Le Global Positioning System (GPS) est un système de géolocalisation mondial. » (français, s.d.)
- Pop-up : des fenêtres intruses qui s'affichent sans avoir été directement sollicité par l'utilisateur.
- Cloud : « Le cloud computing est un modèle qui permet un accès omniprésent, pratique et à la demande à un réseau partagé et à un ensemble de ressources informatiques configurables (comme par exemple : des réseaux, des serveurs, du stockage, des applications et des services) qui peuvent être provisionnées et libérées avec un minimum d'administration. Ce modèle est composé de 5 caractéristiques essentielles, de 3 modèles de services et de 4 modèles de déploiement. » (NIST, 2014)
- SDK : Software Development Kit (Kit de développement)
- Framework : « Un framework désigne en programmation informatique un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application. C'est le framework, ou structure logicielle en français, qui établit les fondations d'un logiciel ou son squelette. » (JDN, 2017)
- SLAM : Simultaneous Localisation and Mapping, permet de bénéficier de la réalité augmentée sans l'utilisation de marqueur spécifique. (Kudan, s.d.)
- LBS : Location-Based Services, est une application logicielle pour les appareils connectés qui ont founit des informations sur notre position.
- CVS : continuous visual search, fonctionnalité de Metaio (Softpedia, 2015)

- HTML : HyperText Markup Language
- CSS : Cascading Style Sheets (Feuilles de style en cascades)
- POI : points of interests, points d'intérêt.
- UDT : User Define Target, technologie propre à Vuforia qui permet à l'utilisateur de l'application de choisir son marqueur pour bénéficier de la réalité augmentée. (Vuforia, s.d.)
- Web-service : Une technologie permettant à des applications de dialoguer à distance via Internet indépendamment des plates-formes et des langages sur lesquels elles reposent. (Dico du web, s.d.)
- API : « Ne laissez pas l'acronyme vous faire peur ! API signifie Application Programming Interface. Le mot le plus important est « interface ». Un exemple commun d'une interface est une télécommande pour votre télévision » (OpenClassrooms, 2017)

1 Introduction

Vous est-il déjà arrivé de pouvoir suivre une course de cyclisme, un marathon, une course de ski de fond et bien plus encore depuis votre tablette ou smartphone ? La réponse à cette question est bien évidemment oui, mais maintenant, poussons cette idée encore plus loin. Vous est-il déjà arrivé de pouvoir regarder ces sports avec le parcours modélisé en trois dimensions et de connaître la position exacte des participants à chaque instant ? Grâce à une évolution certaine dans le domaine de la réalité augmentée cela est désormais possible.

Dans ce travail nous allons principalement parler de réalité augmentée ou AR (Augmented Reality), mais avant toute chose, nous allons retracer l'histoire de la réalité augmentée ainsi que de la réalité virtuelle.

Si vous pensiez que la AR ou la VR sont récent vous vous trompez, en effet l'histoire de la réalité virtuelle débute en 1950 avec la création du premier cinéma immersif par Morton Heilig. Appelé le *Sensorama*, ce cinéma individuel permettait d'exploiter nos différents sens grâce à des ventilateurs et à un siège vibrant qui transportaient l'utilisateur au sein même du film.

En 1968, Ivan Sutherland créa le premier casque de réalité virtuelle et de réalité augmentée. Il était si lourd et inconfortable qu'il fallait le suspendre depuis le plafond avec un bras mécanique, par conséquent on nomma ce casque : l'Épée de Damoclès. Ce dispositif de maintien rendait son utilisation limitée dans les mouvements. Deux capteurs de mouvement (l'un mécanique et l'autre ultrasonique) sont utilisés pour mesurer la position de la tête et ainsi modifier l'image en fonction des mouvements.

Créé en 1982, le gant de données ou *DataGlove*, mesure le déplacement de la main et des doigts et le transmet à l'ordinateur.

Entre 1990 et 2000 les premiers jeux en réalité virtuelle faisaient leurs apparitions avec le Sega VR, un casque qui réagissait aux mouvements de la tête de l'utilisateur. (Realite-Virtuelle, 2014)

De 2000 à 2017 de nombreux casques ont été développés et la réalité virtuelle (ainsi que la réalité augmentée) a été rendu accessible à tous. Comme exemples de casques nous retrouvons l'Oculus Rift, le HTC Vive ou encore le Playstation VR.

Dans tout ça, la réalité augmentée est restée dans l'ombre de sa sœur, la réalité virtuelle. Présenté plus haut, le casque d'Ivan Sutherland était à la fois en réalité virtuelle et réalité augmentée, mais il faut attendre 1996 avec l'introduction des marqueurs 2D, permettant la visualisation d'objets virtuels. Cette avancée importante sera largement diffusée trois ans plus tard par Hirokazu Kato et Mark Billinghurst, au travers de leur plateforme de développement en logiciel libre, AR Toolkit. (Augmented Media, 2011)

Avant de plonger dans le concept de réalité augmentée, nous commencerons par définir le contexte de ce projet et d'expliquer les différences entre les termes de réalité augmentée et de réalité virtuelle en présentant des exemples d'utilisation courante.

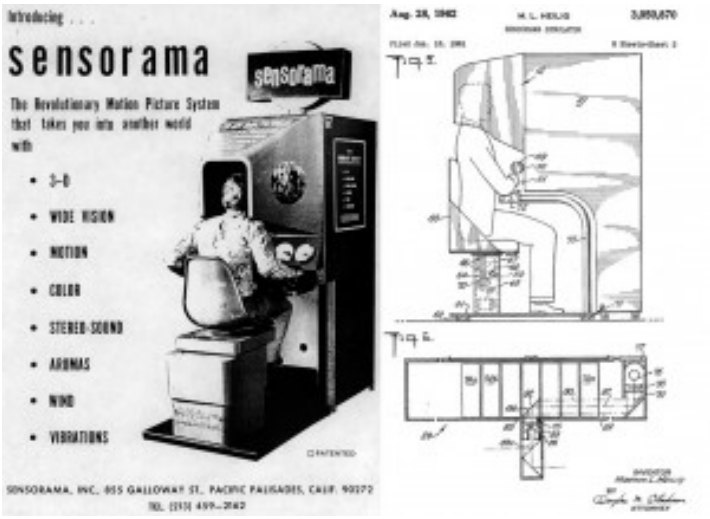
Dans un deuxième temps nous ferons une analyse de l'utilisation de la réalité augmentée dans le milieu sportif et plus particulièrement dans les courses cyclistes.

Dans un troisième temps nous nous attarderons sur les outils les plus populaires et les plus récurrents en matière de réalité augmentée. Nous établirons ensuite un rapide comparatif de ces produits.

Par la suite nous nous intéresserons aux différentes plateformes de développement sous les principaux systèmes d'exploitation utilisés.

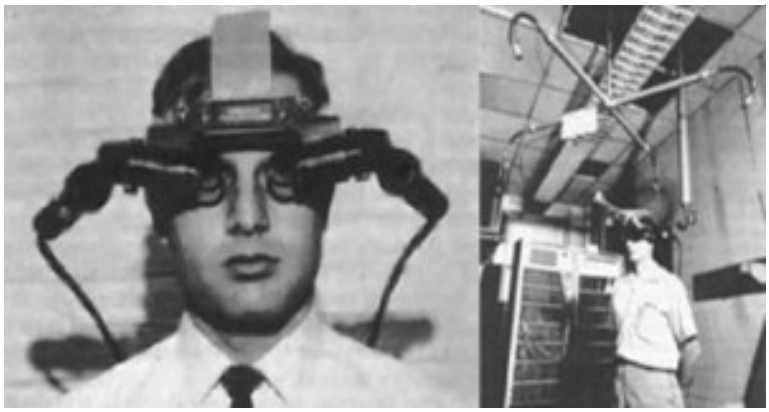
Une fois ces parties décortiquées et définies, nous nous attaquerons à la confection d'un prototype en réalité augmentée qui permettra à l'utilisateur de suivre une course de cyclisme sur son smartphone avec une carte en trois dimensions et de connaître la position exacte des participants en temps réel. Ce projet sera expliqué et détaillé dans la dernière partie de ce document.

Figure 1: Sensorama



Source : http://www.realite-virtuelle.com/lhistoire-realite-virtuelle

Figure 2: Ultimate Display (Épée de Damoclès)



Source : http://www.realite-virtuelle.com/lhistoire-realite-virtuelle

Figure 3: DataGlove

Source : http://www.realite-virtuelle.com/lhistoire-realite-virtuelle

2 Contexte

Ce projet est la conclusion de trois ans en Informatique de Gestion à la HES-So Valais/Wallis de Sierre et permet ainsi l'obtention d'un Bachelor of Science HES-So en Informatique de gestion (Business Information Technology).

L'attribution de ce travail s'est déroulé en trois étapes. Une première étape fut de remplir une feuille avec nos préférences tant en langages de programmation que de programmes. La seconde étape nous a laissé la possibilité de choisir des travaux de Bachelor soumis par les professeurs, pour cela nous avons dû définir une liste comprenant cinq thèmes. Et enfin la dernière étape fut l'attribution du travail à chaque étudiant en fonction de ses listes de préférences et de ses choix de thèmes.

Ce sujet a été mon premier choix lors de la sélection et correspond ainsi pleinement à mes attentes concernant ce travail. Ce projet me permet d'approfondir mes connaissances en développement dans le domaine de la réalité augmentée ainsi que dans la gestion de projet. Il me permet également d'aiguiser mon sens de l'organisation du travail.

Le travail à effectuer est de définir un état de l'art des outils de réalité augmentée, d'analyser ce qu'il se fait dans le domaine sportif en termes de réalité augmentée et plus précisément dans le milieu des courses de cyclisme et en dernier lieu, de créer un prototype utilisant la technologie de réalité augmentée et répondant à un besoin émis par la société Adventures-Lab, basée au Technopôle à Sierre en Valais (Suisse) conjointement avec la société GEO POSITIONING SWISS TECHNOLOGY (GPST).

L'objectif final de ce projet est de pouvoir fournir à Adventures-Lab un prototype fonctionnel qui permet l'affichage sur une carte en trois dimensions des participants à une course cycliste.

3 Réalité augmentée et réalité virtuelle

3.1 Réalité virtuelle

La notion de réalité virtuelle est centrée autour de la perception. Ce n'est pas qu'une simple transposition du réel à l'écran, mais d'une technologie afin de promouvoir une immersion sensorielle totale dans un environnement 100% artificiel, un environnement créé par l'homme et la machine. Un exemple d'utilisation courante est un simulateur de vol, la personne est insérée dans un cockpit qui se veut être une reproduction exacte d'un vrai cockpit d'avion. L'utilisateur est ainsi projeté dans un monde totalement artificiel et sûr afin de se familiariser avec le comportement des instruments et de l'appareil en vol. (rslnmag, 2016)

Figure 4: Simulateur de vol

Source : http://img11.hostingpics.net/pics/319653SIM10.jpg

Vous l'aurez donc compris de part cet exemple que la réalité virtuelle est un environnement créé de toutes pièces par l'homme et la machine. Les masques actuels, tel que l'Oculus Rift pour n'en citer qu'un, proposent également de s'associer avec des éléments intermédiaires tels que des gants ou des manettes pour parfaire l'immersion de l'utilisateur.

Nous avons pu constater que chez certaines personnes, l'utilisation de la réalité virtuelle pouvait provoquer une sorte de mal comparable au mal des transports, appelé Motion Sickness ou cinétose. Ce mal est dû à l'expérience de la vision, du son et/ou du toucher artificiellement créés qui déstabilisent notre cerveau.

3.2 Réalité augmentée

La réalité augmentée, quant à elle, dissocie le réel du virtuel. En effet, il s'agit le plus souvent de rajouter une couche d'informations visiblement artificielle afin d'augmenter le contenu de notre réalité. Ici, au contraire de la réalité virtuelle qui consiste à créer un monde totalement artificiel, nous désirons obtenir plus d'information sur notre propre réalité.

C'est une technologie qui crée une interaction entre une situation réelle et des données virtuelles. Ces données virtuelles sont donc superposées à la réalité grâce à des images ou des modèles en deux dimensions ou trois dimensions via l'intermédiaire de l'écran d'un smartphone, d'une tablette ou encore de lunette connectée. (rslnmag, 2016)

Comme exemple d'appareil et d'application de réalité augmentée, nous pouvons prendre les fameuses Google Glass. Une application permettait, entre autres, grâce aux données GPS de l'utilisateur, de fournir les noms des montagnes, villes, villages, monuments, etc… qui se trouvent en face de son regard. (Google Glass Apps, s.d.)

Figure 5: Google Glass

Source : http://codeaweso.me/presentations/asu-ers-glass/glass.png

Figure 6: exemple application Google Glass

Source : http://www.reussir91.com/sites/default/files/lastermontagne.jpg?1401455217

3.3 Différences AR / VR

Maintenant que les aspects de réalité augmentée et de réalité virtuelle ont été éclaircis, il est beaucoup plus facile de se faire un point de vue sur leurs différences.

En résumé, la réalité virtuelle crée un monde totalement artificiel. Dans un premier temps on pourrait se limiter au visuel grâce aux casques avec leurs écrans mais on nous immerge encore plus dans cet univers avec du son et même le toucher dans certain cas. Avec l'utilisation de capteurs sur l'équipement et sur la personne elle-même, nous sommes capables de se déplacer et d'interagir avec quelqu'un d'autre présent dans ce monde au même moment, bien sûr nous pouvons également voir notre propre corps virtuel. De ce fait, il devient facile de se lancer un objet de l'un à l'autre et de l'attraper sans le faire tomber par exemple.

La réalité augmentée quant à elle, ajoute du contenu digital à notre monde ou réalité pour reprendre ce terme. Avec l'utilisation d'un appareil qui fera l'intermédiaire entre nos yeux et ce que l'on voit, un logiciel ajoutera aux bons endroits le contenu augmenté. Comme appareil il ne faut pas obligatoirement se procurer la dernière paire de Google Glass, en effet, un simple smartphone ou une tablette nous permet de bénéficier de la réalité augmentée. Cette technologie peut aussi utiliser des moyens sonores en plus d'éléments visuels pour rendre l'expérience plus concrète.

3.4 Réalité mixte

La réalité mixte peut être un peu plus compliqué à cerner aux premiers abords mais dans cette section nous allons essayer de l'expliquer au mieux.

Ce principe de réalité veut abolir les frontières entre les mondes virtuels et réels, il s'agit d'intégrer des éléments virtuels sans pour autant démarquer les deux mondes. L'ordinateur holographique autonome HoloLens se prête parfaitement à ce concept, des hologrammes sont projeté sur des verres transparents et autorisent ainsi l'utilisateur de voir et également d'interagir avec ces éléments virtuels.

Comme exemple nous pouvons prendre une simple table en bois carrée. Et sur cette table, avec l'utilisation des HoloLens, nous pouvons faire apparaître une colline avec des arbres et une petite maison au sommet, nous sommes même autorisés à rajouter des interactions, si nous voulons par exemple, modéliser l'environnement de la colline selon nos envies (enlever/rajouter des arbres, des maisons, des fenêtres, etc…). Ou encore un autre exemple peut être de voir la personne en projection holographique devant nous alors que cette dernière se trouve en réalité ailleurs devant un appareil de type Kinect (« Kinect,

contraction de "kinetic" et "connect" désigne un dispositif électronique compatible avec le lecteur blu-ray / console de jeu Xbox One permettant de commander celui-ci vocalement et par gestes à l'aide des caméras et microphones qui le composent. La caméra intégrée au Kinect, peut aussi être utilisée avec l'application Skype, elle est capable de réaliser des vidéos HD (1080p). ») (homecine-compare, s.d.).

Comme dit plus haut, ce concept est un peu plus compliqué à cerner. Ce qu'il faut retenir c'est que la réalité mixte tente d'insérer du contenu virtuel dans notre monde réel sans créer de frontière visible. (rslnmag, 2016)

3.5 Illustrations de dispositifs de AR et VR

Ci-après vous trouverez une sélection de quelques images qui illustrent les différents appareils cités dans les sections des différences AR/VR ainsi que la réalité mixte :

Cette image nous montre qu'il est possible de porter l'ordinateur comme sac à dos et ainsi nous rendre une certaine liberté de mouvement pendant l'utilisation de la réalité virtuelle.

Figure 7: sac à dos VR MSI VR One

Source : http://cdn.windowsreport.com/wp-content/uploads/2016/11/top_vr_backpack_pcs_featured.jpg

Ici, deux exemples de masques de réalité virtuelle, nous trouvons l'Oculus Rift ainsi que le HTC Vive :

Figure 9: Oculus Rift

Source : http://syskb.com/wp-content/uploads/2016/09/oculus-rift-vr-headset-1200x698.jpg

Figure 8: HTC Vive

Source : http://www.etr.fr/articles_images/234469-232332-VR_Web_Product_HMD.png

Maintenant quelques exemples de réalité augmentée, dans cette catégorie nous trouvons le HoloLens de Microsoft :

Figure 10: HoloLens

Source : https://compass-ssl.surface.com/assets/f5/2a/f52a1f76-0640-4a37-a650-51b0902f8427.jpg?n=Buy_Panel_1920.jpg

Un concurrent de Microsoft propose un autre casque de réalité augmentée, dans ce travail nous ne nous attarderons pas à un comparatif des deux car ce projet n'utilise pas ce matériel. Le *Meta 2*, propose aussi de la réalité augmentée mais avec d'autres contraintes que le dispositif proposé par Microsoft. Une image du *Meta 2* :

Figure 11: Meta 2

Source : https://www.metavision.com/assets/product_shot-c0e8b10dbbb42185225ebbd031cd4d993c4e354eed1208c13e318c550b6fc17f.jpg

Dans cet exemple, nous pouvons apercevoir un tapis de jeu, qui, au travers d'une application lancée sur une tablette (ou smartphone) prend vie grâce aux contenus en trois dimensions. Nous pouvons ainsi observer qu'il n'est pas nécessaire d'avoir un équipement très onéreux pour profiter de la réalité augmentée.

Figure 12: AR avec une tablette

Source : https://static.wixstatic.com/media/4d1b95_1206ae55f64d46fc9fce1241b7b0ed18

4 Réalité augmentée dans le milieu sportif

Quand nous parlons de réalité augmentée, nous pensons à une technologie récente dans son développement et son utilisation. Mais aussi surprenant que cela puisse paraître la réalité augmentée a été largement utilisé dans le milieu sportif, notamment par la NBA (National Basketball Association), NFL (National Football League) et MLB (Major League Baseball) afin d'aider au mieux les fans de sports à comprendre les subtilités de certains placements.

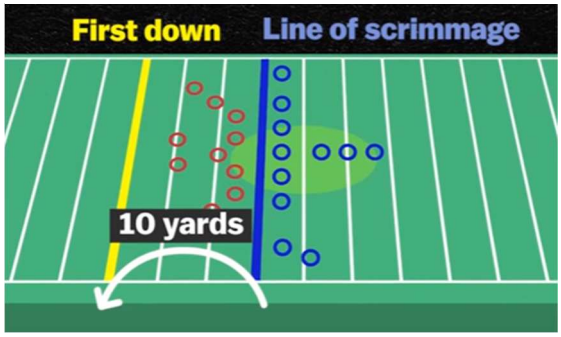
La plupart des gens voient la réalité augmentée comme une technologie du futur mais nous aimons penser que c'est une innovation du présent. Depuis l'arrivée de l'application Pokémon GO, un enthousiasme certain pour la réalité augmentée est né, mais comme dit précédemment cette technologie n'est pas nouvelle. Dans ce chapitre nous allons passer à travers quelques exemples d'utilisation de AR dans le milieu sportif plus précisément.

L'utilisation de la réalité augmentée a été vu pour la première fois par la masse d'une manière très subtile. En effet, en 1998 Sportvision a utilisé le système 1st and Ten Line afin d'ajouter une ligne jaune virtuelle pour connaître la position exacte de la première ligne de Down à atteindre pour l'équipe en possession du ballon, seul l'audience qui regardait le match via leur télévision ont pu observer cet ajout.

Figure 13: NFL Ligne jaune

Source : http://www.huffingtonpost.com/entry/did-sports-really-pave-the-way-for-augmented-reality_us_57b4889be4b03dd53808f61d

Figure 14: Ligne jaune animation



Source : https://www.youtube.com/watch?v=1Oqm6eO6deU

Mais l'exemple ci-dessus n'est pas la première utilisation de la réalité augmentée dans le milieu sportif. En 1996 le système FoxTrax (ou Glowing Puck) a été utilisé par la NHL (National Hockey League), grâce à des senseurs situés dans le palet le téléspectateur pouvait voir apparaître un halo de couleur bleu autour de ce dernier. Quand la vitesse du palet dépassait un certain seuil une trainée rougeâtre remplaçait ce halo bleu. D'une manière générale, ce changement ne fut pas très bien accepté par la communauté des fans

de hockey. Certains y voyaient une amélioration dans le suivie du jeu tandis que beaucoup d'autres tendaient à dire que ce système distrayait le spectateur du jeu en soi et était considéré plus comme un gadget qu'une réelle amélioration.

Quoi qu'il en soit, après le succès de la technologie 1st and Ten Line utilisé par la NFL, ce système fut amélioré par la suite afin d'être le plus stable et réel possible et également d'autoriser son utilisation par la populaire Skycam, la caméra aérienne qui permet une couverture du jeu directement au-dessus du terrain est visible par les spectateurs présents dans le stade.

La NBA utilise également un système similaire au 1st and Ten Line de la NFL. Appelé Virtual 3, cette technologie permet de surligner l'arc des trois points si un joueur tente un tir dans sa portée. Ce visuel disparaît si le tir n'est pas réalisé. Cette couche digitale est faite de telle manière qu'elle ne cache pas les joueurs ou autres informations tout comme la ligne jaune des Downs pour la NFL.

Des expériences en réalité virtuelle ont également été testées. Durant les play-offs de 2016, un partenariat entre Intel et la NBA a été établi. Une vidéo 360 degrés a été créée afin de rejouer les Slam Dunks (smash ou dunk) et autres beaux moments de jeu et permettre ainsi une immersion pour le spectateur presque total. Des idées de diffusion en direct via la réalité virtuelle ont également fleuries auprès des fédérations de sport. (huffingtonpost, 2016)

En 2010, IBM a lancé une application en lien avec le tournoi de Tennis de Wimbledon. Elle permettait à l'utilisateur de s'orienter plus facilement sur le lieu du tournoi grâce aux données de géolocalisation. Ce logiciel pouvait nous renseigner sur une multitude de choses, de la longueur de la file d'attente pour le Champagne et les fraises à la crème au distributeur de billet de banque le plus proche. Cette application nous donnait aussi la possibilité de savoir le résultat du tournoi et même de voir des actions en direct des matchs. IBM a donc sorti une application qui permettait à l'utilisateur de « voir » à travers les murs. (The Telegraph, 2010)

5 Réalité augmentée et le cyclisme

Pour revenir sur le contexte de ce travail, qui est de fournir une application en réalité augmentée permettant à l'utilisateur de suivre une course de cyclisme avec une autre perspective, nous nous rendons très vite compte qu'à l'heure actuelle il y'a peu, voir aucune application pour le spectateur. Par contre une pléthore de dispositifs existent et sont en développement afin de soutenir l'athlète, qu'il soit professionnel ou amateur, dans son entrainement ou son expérience sur la route.

Nous allons rapidement passer en revue quelques dispositifs qui sont destinés aux coureurs eux-mêmes sans entrer trop dans les détails de chacun.

Les Raptor AR Glasses de Everysight Company est une lunette utilisant les technologies de réalité augmentée. Elle propose l'affichage constant des informations importantes pour l'utilisateur comme la vitesse, la cadence, la fréquence cardiaque et d'autres informations. Avec sa caméra haute définition incluse dans les lunettes elle permet de réaliser des photos et vidéos facilement du point de vue du cycliste. Connecté à notre smartphone elle offre aussi la possibilité de voir les notifications des messages ou des réseaux sociaux entrant. Une fonctionnalité de navigation est incluse et permet de connaître la route tel un GPS. (SportsWearable, 2016)

Figure 15: Raptor Glasses

Source : https://everysight.com/

Figure 16: exemple de vue des Raptor

Source : https://www.youtube.com/watch?v=96Iz7w-uOog

Un second exemple sont, toujours dans le même registre, les lunettes de réalité augmentée solos™. Elles ont été créées conjointement avec l'équipe de cyclisme des USA et ont pour but de répondre au mieux aux besoins des athlètes. Les fonctionnalités diffèrent peu du modèle Raptor dans l'ensemble mais le design et l'affichage des informations n'est pas pareil. Ci-dessous une illustration des lunettes solos™ : (Solos, s.d.)

Figure 17: solos™

Source : https://www.youtube.com/watch?v=96Iz7w-uOog

Figure 18: exemple de vue solos™

Source : https://www.kickstarter.com/projects/1101608300/solos-smart-cycling-glasses-with-heads-up-micro-di

6 Apports de la réalité augmentée dans les courses de cyclisme

6.1 Introduction

Dans ce chapitre nous allons essayer de lister les apports bénéfiques de la réalité augmentée pour les courses de cyclisme, non pas pour les athlètes mais pour les spectateurs.

Notre application vise tant les téléspectateurs que les fans postés en bordure de route afin de pouvoir apercevoir les athlètes l'espace de quelques secondes. À la télévision nous pouvons avoir différentes prises de vue des caméras mais diffusaient dans l'ordre choisi par le régisseur, le téléspectateur n'a aucune influence sur les images ou les prises de vue, il est uniquement observateur passif de ce qui se déroule sous ses yeux. L'effet de perspective est également délicat dans le sens où une vue aérienne, par exemple, rend difficile la représentation de l'effort à fournir de la part des athlètes dans une cote.

Grâce à la réalité augmentée il nous est possible d'afficher le parcours, la carte, en trois dimensions. Ce nouvel élément visuel nous autorise à naviguer comme bon nous semble autour de la carte et ainsi avoir une perspective en relief du circuit. Nous pouvons placer les coureurs à leur position actuel sur la carte et même ajouter des interactions. Nous avons la possibilité de mettre en place des filtres qui nous permettent de choisir l'affichage d'un ou plusieurs athlètes, ou tout le monde et cela à choix de l'utilisateur. Bien sûr il n'y pas de représentation réelle du participant, c'est-à-dire que nous n'allons pas voir le cycliste en soit sur la carte mais une modélisation de lui qui peut être représentée par tout ce que désire le développeur et peut-être même laisser la possibilité du choix de l'illustration par l'utilisateur.

La réalité augmentée ne donnera certainement pas plus d'informations concernant les cyclistes que la télévision mais apportera une manière différente de suivre la course, une autre perspective qui leur apportera plus de liberté d'interaction avec le média de diffusion.

6.2 Spécificités

Nous allons tenter de lister les apports bénéfiques de la réalité augmentée par rapport à un média classique et par conséquent l'intérêt de l'implémentation et de l'utilisation de cette technologie.

- Visualisation du terrain en trois dimensions

  Ce point est certainement le plus important, il nous offre une toute nouvelle approche du terrain et nous rend sensible à la dénivellation. Nous pouvons aisément concevoir l'effort que dois fournir un cycliste sur une pente raide ou un col. L'affichage en trois dimensions ajoute également un effet d'immersion pour l'utilisateur.

- Filtrer les informations

  Le filtrage d'information n'a pas vraiment de lien direct avec la réalité augmentée car dans une application standard nous sommes aussi à même de réaliser une telle action si les développeurs l'ont intégré. Il est toutefois intéressant de le souligner car nous pouvons instaurer des filtres pour voir un seul compétiteur ou un groupe défini (par position ou par équipe). Nous pouvons également insérer des filtres pour contrôler l'environnement, c'est-à-dire avoir la possibilité d'afficher ou non des éléments en trois dimensions sur la carte.

- Ajout de contenu publicitaire

  Cet ajout doit se faire de façon subtile, car l'utilisateur est généralement réticent à l'idée d'avoir une application contenant trop de publicité. Nous allons revenir encore une fois sur la carte en trois dimensions, car vous l'aurez compris maintenant, c'est cette dernière qui apporte la plus grande innovation et intérêt dans l'application. Comme les villes, forêts, champs, lacs, bref tout ce qui entourent la course devraient être modélisée en trois dimensions (pas obligatoirement représentatif de la réalité car modéliser une ville entière prend énormément de temps et de ressources), il est aisé d'insérer dans tout ce petit monde des petits « pop-up » qui signaleront l'emplacement d'un lieu important, d'un restaurant, d'un musée, d'un commerce, etc…

  L'expérience de l'utilisateur n'est pas diminuée par l'ajout de ces points d'intérêts s'ils sont ajoutés de manière non envahissante.

6.3 Conclusion

Notre application se distinguera par son contenu en trois dimensions et des informations disponibles pour chaque participant à la course cycliste. L'accès aux informations doit se faire le plus simplement possible pour la personne se servant de l'application. Les petites publicités devront se fondre subtilement dans le décor afin de ne pas perturber l'expérience de l'utilisateur. Il s'agira principalement de mettre en évidence des lieux et espaces culturels mais des publicités pour des commerces privés peuvent également en faire partie et contribuent ainsi à la rentabilité du logiciel.

7 Outils de réalité augmentée

7.1 Introduction

Avant de commencer à détailler notre prototype, nous allons nous pencher sur les outils de réalité augmentée ainsi que sur leur environnement de développement. Ce chapitre va donc énumérer un certain nombre de moyens offrant la technologie de réalité augmentée. Nous allons analyser chaque outil et définir leurs spécificités ainsi que les systèmes d'exploitation touchés, soit Android, iOS et Windows.

Il s'agit ici d'une liste représentant une petite partie des outils AR présent sur le marché. Ils ont été choisis pour analyse car ils sont le plus souvent cités sur des sites proposant un choix de dispositifs de réalité augmentée.

En deuxième partie nous ferons un bref résumé des environnements de développement (IDE) qui nous permettent de développer des applications mobiles et qui intègrent également les outils de réalité augmentée.

En conclusion, nous vous présenterons l'IDE ainsi que l'outil AR choisis et pourquoi ils nous ont séduit.

7.2 Outils AR

7.2.1 ARPA

Le SDK de ARPA nous est fourni par la société Arpa Solutions. Ce SDK permet la détection d'image (seul ou multiple), l'affichage d'objets en 3 dimensions en temps réel ainsi que des interactions de l'utilisateur sur l'objet en question comme la sélection, la rotation ou encore définir sa taille, sont là quelques fonctionnalités à disposition. Des compléments au SDK de base peuvent être apportés comme le ARPA GPS SDK qui offre la possibilité d'utiliser les coordonnées GPS afin de bénéficier de plus amples informations sur l'environnement autour de soi. Le ARPA GLASS SDK et le ARPA Unity Plugin permettent quant à eux le développement d'applications avec les Google Glass ainsi que le moteur de jeux Unity.

Malheureusement, depuis fin 2015, le nom de domaine arpa-solutions.net semble avoir expiré (developereconomics, 2015) et après quelques recherches, nous n'avons pas trouvé d'endroit ou télécharger le SDK de ARPA Solutions.

7.2.2 ARLab

ARLab est une licence uniquement commerciale, il n'y a pas de version d'essai gratuite. Elle offre deux types de produits, le AR Browser SDK, qui permet d'ajouter ou de supprimer des points d'intérêts d'une scène en temps réel, interagir avec ces derniers comme par exemple les toucher ou pointer la camera vers eux et de réaliser une action comme envoyer un SMS ou encore partager sur Facebook les informations que l'application nous affiche. Le deuxième type de produit est la correspondance d'image ou image matching. Comme son nom l'indique il permet la reconnaissance d'image. Leurs points forts sont :

- La reconnaissance d'image en temps réel
- La détection de plusieurs images en même temps
- Des milliers d'images sont supportées dans des groupements de 50 à 60 images
- La détection des QR code
- Fonctionne sans connexion internet
- Réponse rapide grâce à l'optimisation des composants matériels (hardware)

Comme dit plus haut, seule une version commerciale est proposée à un prix de 199€ pour le AR Browser et de 299€ pour Image Matching. Ce prix est proposé par application développé. Ce SDK offre la possibilité de créer des applications pour Android et iOS. (ARLab, s.d.)

7.2.3 DroidAR

DroidAR est un Framework en open-source, c'est-à-dire que chaque personne a la possibilité de le redistribuer librement et d'accéder au code source du programme. Ce Framework est utilisé avec l'outil de développement Eclipse. Seul des applications pour Android peuvent être créées avec DroidAR. Les mouvements de l'utilisateur peuvent être perçus par l'application, l'affichage et l'interaction avec des objets en trois dimensions (par exemple cliquer dessus) ainsi que la détection de marqueur sont une partie des possibilités offertes par ce Framework.

Il est disponible uniquement pour les applications fonctionnant sous Android. (DroidAR, s.d.)

7.2.4 Metaio

Il n'est malheureusement plus possible d'acheter une licence pour Metaio depuis le 15 décembre 2015, leur site annonce que les produits et souscriptions ne sont plus accessible mais que les licences courantes continueront jusqu'à expiration. (techcrunch, 2015) Néanmoins, le SDK est toujours libre d'accès sur Softpedia mais avec les restrictions d'une licence gratuite. (Softpedia, 2015)

Cet outil convient pour les types d'appareils suivants :

- Android
- iOS
- Windows

Le SDK de chez Metaio peut être utilisé sur différentes plateformes, spécifiques pour chaque système d'exploitation cité ci-dessus, en utilisant les langages de programmation Java (pour Android), Obj-C (pour iOS) et C++ (pour Windows). L'installation de ce dernier est facilité par le téléchargement automatique de tous les fichiers nécessaires pour la capture, l'affichage et l'utilisation des senseurs.

Comme beaucoup d'autres outils de AR, Metaio propose les services suivants d'utilisation pour la réalité augmentée (metaio, s.d.) :

- Images en 2 dimensions
- Objets et environnements en 3 dimensions
- SLAM
- LBS
- CVS

7.2.5 Vuforia

Le SDK de Vuforia permet de développer sur Android, iOS et également pour les applications Windows depuis la version 6 de Vuforia en incluant les tablettes/ordinateurs de la gamme Surface de Microsoft ainsi que les HoloLens et les applications Windows 10.

Certaines applications doivent reconnaître quelques images tandis que d'autres doivent en distinguer des centaines, c'est pourquoi Vuforia a mis en place deux manières de stocker les images :

- Base de données dans l'appareil

  Cette option propose de sauvegarder directement dans l'appareil via l'application les images nécessaires à la détection. Cette technique est plus rapide mais ne peut contenir que quelques images.

- Base de données sur un cloud

  En choisissant de stocker les images sur une base de données en cloud, c'est-à-dire sur un serveur externe en utilisant internet, il est possible d'avoir une masse d'images de détection bien plus conséquente qu'en prenant l'option de stockage sur l'appareil. Le bémol est que l'accès à ce service distant peut prendre un certain temps et donc ralentir notre application.

Pour déterminer quelle option choisir entre une base de données sur un appareil ou un cloud il faut bien définir les besoins de l'application. Si nous n'avons que quelques images à utiliser comme cible de détection il sera préférable d'opter pour une base de données local tandis qu'à l'inverse, si nous devons avoir une multitude d'images de détection, une approche de base de données sur un serveur distant serait plus logique.

La création et la gestion des images cibles est très facile en utilisant Vuforia, en effet il nous suffit de se connecter à leur plateforme web et d'y charger l'image ou l'objet voulu. En fonction du logiciel de développement utilisé, il suffira de télécharger un paquet spécifique et de l'installer dans notre projet pour être reconnu et utilisable.

Ce SDK offre aussi la possibilité d'utiliser des objets en trois dimensions comme cible de détection. Une option de Smart Terrain ainsi qu'un paquet dit User Defined Target autorise l'utilisateur à définir en temps réel son image ou son objet cible, ce n'est pas une technologie dite SLAM mais elle peut faire office de substitue. (Vuforia, s.d.)

7.2.6 Wikitude

Le SDK de Wikitude est l'un des plus complet en matière de technologie de réalité augmentée. Il propose en effet le Instant Tracking, avec cette option il n'est plus nécessaire d'avoir un marqueur spécifique. Il scanne notre environnement, autant à l'intérieur qu'à l'extérieur, et autorise ainsi l'ajout de contenu augmenté par-dessus notre réalité. Le Extended Tracking rejoint le Instant Tracking dans le sens où il faut simplement détecter une fois notre marqueur et nous pouvons ensuite bouger librement sans garder l'image cible dans le focus de la caméra.

Comme d'autres SDK, Wikitude propose la reconnaissance d'image ainsi que celle d'objet en trois dimensions. Il propose également le même service que le précédent SDK, Vuforia, en laissant le choix au développeur d'enregistrer les marqueurs soit directement en local sur l'appareil soit sur un serveur distant (cloud).

Nous pouvons également utiliser les données de géolocalisation afin de bénéficier de la réalité augmentée. Par rapport à nos données GPS, l'appareil est capable d'ajouter des informations digitales au travers de la caméra. Un exemple de cette application est la reconnaissance des montagnes, villes, villages autour de nous.

Le SDK de Wikitude peut être utilisé avec différentes plateformes de développement et permet l'exportation de ses applications sur presque tous les systèmes d'exploitation.

Les plateformes de développement et les systèmes d'exploitation sont les suivants :

- Android
- iOS
- Smart Glass
- Unity 3D
- CORDOVA/PHONEGAP
- APPCELERATOR TITANIUM
- XAMARIN

Pour le moment, Wikitude ne propose pas de solution pour les appareils utilisant Windows (UWP) mais ils sont actuellement en développement du SDK permettant de le faire. (Wikitude, s.d.)

7.2.7 ARToolkit

ARToolkit est un outil de réalité augmentée basique, il permet la reconnaissance d'images, de marqueurs, de code-barres mais pas d'objets en trois dimensions contrairement à d'autres SDK.

Malgré une technologie qui peut paraître un peu faible, ARToolkit se distingue par le fait qu'il nous autorise à développer sur presque toutes les plateformes :

- OS X
- Windows
- iOS
- Android
- Linux 32/64-bit
- Unity

ARToolkit est la première plateforme logicielle permettant le développement en réalité augmentée. Il fut lancé en 1999 et est l'un des outils les plus utilisé. En dépit de son histoire, de sa popularité et de sa gratuité, la documentation de développement est très limitée. Elle inclut des tests d'application mais tous ne peuvent pas être facilement construits. La version ARToolkit 6 est actuellement en développement et nous pouvons télécharger la béta.

Pour résumé, ARToolkit est un outil gratuit et open source mais quelque peu limitée dans son emploi et sa technologie comparé à d'autres outils de réalité augmentée mais il touche une large gamme de plateformes et systèmes d'exploitation. (ARToolKIt, s.d.)

7.2.8 LayAR

Les SDK de chez LayAR propose différentes technologies, la première est la Layar Vision. Elle utilise la détection et le suivi pour augmenter des images. Les marqueurs cibles sont stockés dans l'appareil pour une réactivité rapide.

Le HTML widget est une des autres technologies. Nous pouvons rajouter du contenu augmenté (vidéos, images, sons, etc…) dans nos pages HTML directement grâce à du code HTML, JavaScript et CSS.

Geo Layer, il permet la localisation de l'utilisateur et ainsi de renvoyer du contenue augmentée ou encore des points d'intérêts (POI).

Au lieu d'images comme marqueurs nous pouvons utiliser des objets en 3 dimensions.

LayAR est payant mais propose un essai gratuit sans limitation de 30 jours. Au-delà de ce temps, si aucune solution de paiement n'a été faite avec la société LayAR, tous documents et tous projets contenus dans le SDK seront détruits. Le système de paiement est différent des autres outils, qui eux proposent un prix soit par application soit annuel. LayAR donne un prix par page, nous pouvons utiliser leur plateforme de développement graphique ou encore combiné le SDK avec Android Studio ou iPhone et iOS. Il est également possible de l'installer sur la plateforme de développement Eclipse mais cette option est devenue obsolète et il est préférable de l'utiliser avec Android Studio. Il est également possible de l'employer avec PhoneGap, c'est un logiciel qui utilise comme Framework Apache Cordova (pour rappel, Apache Cordova est une plateforme qui permet la création d'application pour Android et iOS avec comme langage le HTML, CSS et JavaScript).

Malheureusement très peu, et même aucune information, n'a été trouvé concernant la création d'application avec les appareils utilisant Windows et également le développement via des moteurs de jeux comme Unity. (LayAR, s.d.)

7.2.9 Kudan

Kudan se définit en pensant avoir la meilleure technologie SLAM sur le marché de la réalité augmentée. Pour rappel, SLAM permet la localisation dans notre espace afin de pouvoir ajouter du contenu augmenté, il nous rend possible l'ajout en temps réel d'objet en trois dimensions dans notre espace de vie, par exemple nous pouvons ajouter un fauteuil dans un coin de notre pièce avec les dimensions souhaitées et ainsi avoir un aperçu d'un vrai siège à cet emplacement précis.

Leur SDK est utilisable sur la plupart des plateformes de développement :

- Android
- iOS
- Unity
- Windows (via Unity)

En plus de la technologie SLAM, Kudan met à disposition la reconnaissance de marqueur spécifique en deux ou trois dimensions. Une configuration avancée autorise l'utilisation de la haute définition (HD) de notre caméra ainsi que la haute qualité du rendu des textures des modèles en trois dimensions en temps réel. Les options de plusieurs marqueurs ou encore la détection étendue ne sont pas un problème pour le SDK de Kudan.

Le code de KudanCV est écrit en C++ et possède une architecture spécialement optimisée écrite en assembleur, ce qui lui donne une vitesse et une robustesse de performance avec un minimum d'impact sur la mémoire. Ces spécificités rendent donc le SDK de Kudan très intéressant s'il faut allier performance et haute qualité dans notre application.

L'utilisation de Kudan au sein de Unity peut se comparer au SDK de Vuforia. Il nous faut également passer par le site internet dans la section développeur afin d'y ajouter une clef de licence ainsi qu'un ou plusieurs marqueurs (si besoin de ces derniers). (Kudan, s.d.)

7.3 Comparatif des outils de AR

Ce tableau nous offre un petit récapitulatif visuel de chaque outil de réalité augmentée. Nous pouvons observer les plateformes supportées ainsi que le moteur de jeux Unity.

Tableau 1: comparaison des outils AR

Outils	Android	iOS	UWP	Unity 3D	Remarques
ARPA	✓	✓	✗	✓	N'existe plus
ARLab	✓	✓	✗	?	Pas de version gratuite
DroidAR	✓	✗	✗	✗	Uniquement sur Android
Metaio	✓	✓	✓	✓	License plus disponible, racheter par Apple Inc.
Vuforia	✓	✓	✓	✓	Utilisation gratuite
Wikitude	✓	✓	✗	✓	Payant avec une licence d'essai
ARToolkit	✓	✓	✓	✓	Gratuit et open source mais limité
LayAR	✓	✓	✗	?	License d'essai de 30 jours
Kudan	✓	✓	✓	✓	Utilisation gratuite

Source : personnelle

8 Moteurs de jeux et logiciels

8.1 Android

8.1.1 Android Studio

Android Studio est l'environnement de développement intégré officiel de Android. Développer par JetBrains, ce logiciel nous permet de créer des applications pour le système d'exploitation Android en utilisant Java ainsi que le nouveau langage de programmation Kotlin.

Un émulateur est intégré à l'IDE et nous donne ainsi la possibilité de tester directement notre application sans avoir besoin de générer un APK (format de fichier Android, Android Package). Une multitude de format peut être utilisé pour simuler l'environnement dans lequel notre application évoluera, du smartphone à la tablette avec des résolutions et orientations d'écran différent. (Android Studio, s.d.)

8.2 iOS

8.2.1 XCode

XCode permet le développement d'application pour Mac, iPhone, iPad, Apple Watch et Apple TV, bref tous les appareils de chez Apple. Cette IDE utilise les Framework Cocoa et Cocoa Touch. Un bouton permet de scinder l'écran de travail en deux parties, une partie avec notre code actuelle et la seconde partie affiche automatiquement des fichiers utiles en lien avec celui ouvert. Par exemple si nous avons le fichier MyClass.m ouvert dans la première fenêtre, l'autre nous proposera le fichier MyClass.h.

Un constructeur d'interface est également disponible, il permet de créer rapidement et simplement l'interface visible de notre application. L'intégration du code source avec le visuel peut se faire en même temps. (Apple, s.d.)

8.2.2 AppCode

AppCode est un environnement de développement mis en place par JetBrains tout comme Android Studio. Il se distingue par une navigation au sein de notre projet très simple et efficace. Tout comme XCode, il propose de créer des applications uniquement pour les produits de chez Apple.

AppCode nous offre deux sortes de complétion de code, une complétion basique basée sur le code que l'on est en train d'écrire ou une approche plus intelligente en faisant des propositions plus rigoureuses et offre un filtre de suggestion plus précis. (JetBrains, s.d.)

8.3 UWP

8.3.1 Visual Studio 2017

Visual Studio est un produit développé par Microsoft. Il offre la possibilité de développer énormément de type d'applications différentes, développement web, cloud, mobile, multiplateforme, bureautique, etc...

Visual Studio propose comme langage de programmation le C#, VB.NET, F#, C++, Python, Typescript, JavaScript, HTML et bien d'autre encore.

Nous pouvons faire du multiplateforme grâce à des APIs natives, nous avons la faculté de développer des applications pour Android, iOS et Windows. (Microsoft, s.d.)

8.4 Unity Engine

8.4.1 Unity

Unity est un moteur de jeux. Il permet de créer des jeux vidéo en deux ou trois dimensions. Toute la partie Scripting se fait au travers d'un outil externe, soit MonoDevelop soit Visual Studio.

Unity offre la création d'un monde virtuel où l'on peut ajouter, enlever, modifier à notre guise des éléments. Couplé avec l'un des outils de réalité augmentée compatible avec Unity, nous pouvons ajouter du contenu digital sur notre propre monde avec l'aide de marqueur. Ces derniers agissent comme un point de repère pour le logiciel afin qu'il sache où son univers virtuel commence et ainsi pouvoir placer les éléments.

Unity est un moteur de jeux des plus rependu avec une très grande communauté de développeurs. Son utilisation est gratuite mais quelques fonctionnalités sont réservées aux utilisateurs payant une licence (comme les outils analytiques).

Les langages de Scripting utilisés par Unity sont le C# ainsi que le JavaScript. (Unity, s.d.)

8.5 Unreal Engine

8.5.1 Unreal Engine 4

Unreal Engine est au même titre que Unity, un moteur de jeux. Il permet également de créer un monde virtuel en deux ou trois dimensions. A l'inverse de Unity, Unreal Engine utilise comme langage de programmation le C++ ou encore BluePrints.

BluePrints est un système de visual scripting, il a été créé en particulier pour les designers afin qu'ils puissent en toute simplicité faire leurs produits sans avoir besoin d'un développeur. Mais un développeur ne peut être remplacé par BluePrints car il faudra tout de même écrire des scripts afin de perfectionner son jeu ou son application.

Unreal Engine et Unity sont les moteurs de jeux les plus repandus, autant chez les indépendants que chez les gros éditeurs. Ci-dessous une petite liste de jeux populaires crée avec Unity et Unreal Engine :

- Pokemon GO (Unity)
- Temple Run (Unity)
- Rust (Unity)
- Rocket League (Unreal Engine)
- Batman : Arkham series (Unreal Engine)
- Borderlands 1 et 2 (Unreal Engine)

(Unreal Engine, s.d.)

8.6 Unity et Vuforia

Pour créer notre prototype nous allons utiliser comme moteur de jeux Unity et comme outil de réalité augmentée Vuforia.

Pourquoi Unity ?

Unity utilise comme langage le C# ainsi que le JavaScript, ces deux langages ont été abordés durant notre formation à la HES-So Valais/Wallis. Un poids en moins car apprendre un nouveau langage prend un certain temps et ce travail n'en dispose pas d'une quantité illimitée. La communauté de Unity est extrêmement grande, il y'a des forums, vidéos, tutoriels pour presque tout. Il est simple et rapide de trouver une aide concernant du code ou de l'utilisation du logiciel en lui-même. Unity est gratuit en dessous de 100'000 dollars US de chiffre d'affaires annuel.

Pourquoi Vuforia ?

Un premier critère est que l'outil de réalité augmentée touche tous les systèmes d'exploitation soit Android, iOS ainsi que Windows et qu'il soit disponible sur Unity. Avec ce premier critère nous avons pu éliminer une bonne partie des outils. Seul reste Metaio, Vuforia, ARToolKit et Kudan.

Un autre critère est la gratuité dans une certaine mesure. Si le prototype séduit, son développement sera repris et amélioré et par conséquent la société responsable disposera d'une licence payante de l'outil. Ce dernier point nous pose une nouvelle condition qui est d'utiliser le même outil de réalité augmentée que l'entreprise qui reprendra le développement et il en va de même pour le moteur de jeux utilisé. Adventures-Lab s'aidant de Unity et de Vuforia le cas peut être clair, mais nous aurions très bien pu utiliser Unity avec un autre outil de réalité augmentée proposant peut-être de meilleures dispositions pour notre prototype. Mais Vuforia remplit son rôle parfaitement et il n'est nullement nécessaire d'utiliser une autre technologie.

Nous allons tout de même filtrer nos quatre derniers outils, voici leur spécificité :

- Metaio

  Une licence payante n'est plus disponible pour le moment mais une version gratuite du Framework est toujours en circulation.
- ARToolKit

  Bon outil, il est l'un des premiers outils de réalité augmentée sur le marché mais il est faible comparé aux autres en termes d'utilisation.
- Vuforia

  Outil totalement gratuit et non limité, hormis un filigrane présent dans l'application sauf avec une licence payante. En dessous de 100'000 dollars US de chiffre d'affaires annuel la licence est gratuite.
- Kudan

  Du même niveau que Vuforia mais avec une technologie SLAM supérieure. Vuforia n'a pas de SLAM à proprement parler mais le remplace avec un mode de détection étendue. Utilisation gratuite également avec possibilité de licence payante.

Nous pouvons donc réduire notre liste à deux outils, Vuforia et Kudan. Vuforia reste cependant notre choix car la société Adventures-Lab l'utilise et il remplit ses fonctions.

9 Prototype

9.1 Introduction

Nous allons donc passer à la partie traitant du développement de notre prototype. Ce dernier a pour but, nous vous le rappelons, de permettre à l'utilisateur de voir une course cycliste en direct via la technologie de réalité augmentée sur son smartphone ou sa tablette.

Les outils nécessaires au développement de l'application étant définis, nous pouvons entrer dans le vif du sujet. La gestion de ce prototype a été faite avec la méthodologie Scrum, un product backlog, un fichier qui contient toutes les fonctionnalités, a été réalisé afin de définir les besoins et attentes de chaque partie et le travail a été effectué en sprint, c'est-à-dire que chaque point (ou fonctionnalité) du product backlog, appelé use case, ont été développé et implémenté dans un temps imparti.

Le prototype va être présenté de la manière suivante, pour chaque use case, le code s'y référant ainsi que son utilisation sera présentée. Des figures viendront illustrer graphiquement les dires tout au long des explications.

9.2 Product Backlog

9.2.1 Use cases

9.2.1.1 Créer les différentes classes objets

Premier élément de notre product backlog, ce use case se définit ainsi :

« En tant que développeur/Responsable de projet, j'aimerai pouvoir créer des classes objets du type voulu afin d'avoir toutes les informations nécessaires à chaque participant dans un objet. »

Nous avons donc créé une classe objet pour chaque niveau. Tout d'abord nous allons vous montrer un échantillon de données fourni par GPST.

Figure 19: échantillon de données GPST


```
{
    "timestamp":29040,
    "groups":[
        {
            "distance":0,
            "gap":"+0:00",
            "athletes":[
                {
                    "hr":"",
                    "pow":"",
                    "bib":301,
                    "rank":1,
                    "IRM":"",
                    "gname":"SYLVAIN",
                    "lat":46.3604306667,
                    "nationality":"SUI",
                    "spd":"0.00 kmh",
                    "lon":6.2049955,
                    "valid":1,
                    "fname":"BRUNNER",
                    "team":"U17",
                    "cad":""
                }
```


Source : personnelle

Le bloc de données ci-dessus représente seulement le début d'un fichier complet, la suite se compose des autres objet « athletes ». Il est réceptionné en format JSON. Tout d'abord nous pouvons observer trois objets distincts :

En jaune, le « timestamp » est une référence du temps en seconde depuis minuit UTC et contient également une collection de groupes (« groups »).

En vert, c'est les informations du groupe (« groups »). La distance entre les groupes s'il y'en a plusieurs. Le « gap » représente le temps qui sépare les groupes du premier groupe et enfin il contient une collection d'athlètes.

En bleu nous avons donc nos athlètes avec les références suivantes :

- hr: rythme cardiaque
- pow: puissance
- bib : numéro de dossard
- rank : rang
- IRM : statut de l'athlète
- gname : prénom
- lat : latitude
- nationality : nationalité
- spd : vitesse
- lon : longitude
- valid : validité de la réception GPS
- fname : nom de famille
- team : l'équipe
- cad : cadence

Maintenant voici nos classes objets utilisées dans nos scripts Unity, ces derniers vont être instanciés, c'est-à-dire alimenter ou créer, en fonction des données reçues par le fichier JSON. Nos objets ont été défini avec la même architecture que celle du fichier JSON :

Figure 20: objet Timestamp

```
public class TimeStamp
{
    public int timeStamp { get; set; }
    public List<Groups> groups { get; set; }
```

Source : personnelle

Notre premier objet fait référence à la partie en jaune de l'échantillon de données présenté plus haut. Nous retrouvons une variable qui contiendra le temps ainsi qu'une collection d'objet de type « Groups ».

Figure 21: objet Groups

```
public class Groups
{
    public int distance { get; set; }
    public string gap { get; set; }
    public List<Athletes> athletes { get; set; }
    public int no { get; set; }
```

Source : personnelle

Le second objet montre la partie verte, ces variables prennent en compte la distance et le « gap ». Nous pouvons également apercevoir une troisième variable qui n'est pas présente dans notre échantillon, cette dernière fait référence au numéro du groupe.

Figure 22: objet Athletes

```
public class Athletes
{
    public string hr { get; set; }
    public string pow { get; set; }
    public int bib { get; set; }
    public int rank { get; set; }
    public string IRM { get; set; }
    public string gname { get; set; }
    public double lat { get; set; }
    public string nationality { get; set; }
    public string spd { get; set; }
    public double lon { get; set; }
    public int valid { get; set; }
    public string fname { get; set; }
    public string team { get; set; }
    public string cad { get; set; }
```

Source : personnelle

Et voici notre dernier objet, l'encadrer bleu. Nous pouvons distinguer toutes les variables représentant les mêmes caractéristiques que les données entrantes.

9.2.1.2 Instancier les classes objets

Maintenant que nos classes objets ont été créées, il nous faut les alimenter avec le jeu de données reçu par GPST en format JSON. Nous allons donc effectuer une opération qui va transformer le format JSON en objet C# utilisable, nous appelons ça le parsing (analyse syntaxique).

Pour pouvoir « parser » nos données, il nous faut créer des parseurs spécifiques car nous utilisons la technologie LitJSON, et cette dernière ne comporte pas de parseurs. Nous devons tout d'abord créer un objet qui contiendra toutes les informations du JSON pour ensuite utiliser un parseur qui lui va segmenter les informations dans les objets voulus. Ci-dessous, nous vous illustrons notre parseur pour les objets athlètes :

Figure 23: parseur athletes

```
new Athletes(itemData["groups"][j]["athletes"][i]["hr"].ToString(),
             itemData["groups"][j]["athletes"][i]["pow"].ToString(),
             (int)itemData["groups"][j]["athletes"][i]["bib"],
             (int)itemData["groups"][j]["athletes"][i]["rank"],
             itemData["groups"][j]["athletes"][i]["IRM"].ToString(),
             itemData["groups"][j]["athletes"][i]["gname"].ToString(),
             (double)itemData["groups"][j]["athletes"][i]["lat"],
             itemData["groups"][j]["athletes"][i]["nationality"].ToString(),
             itemData["groups"][j]["athletes"][i]["spd"].ToString(),
             (double)itemData["groups"][j]["athletes"][i]["lon"],
             (int)itemData["groups"][j]["athletes"][i]["valid"],
             itemData["groups"][j]["athletes"][i]["fname"].ToString(),
             itemData["groups"][j]["athletes"][i]["team"].ToString(),
             itemData["groups"][j]["athletes"][i]["cad"].ToString()));
```

Source : personnelle

Il s'agit bien sûr d'une partie du parseur à titre d'illustration. Mais grâce à ce bout de code nous sommes maintenant en mesure d'utiliser les informations que cet objet contient dans notre application.

9.2.1.3 Création de la carte 3D et implémentation dans Unity

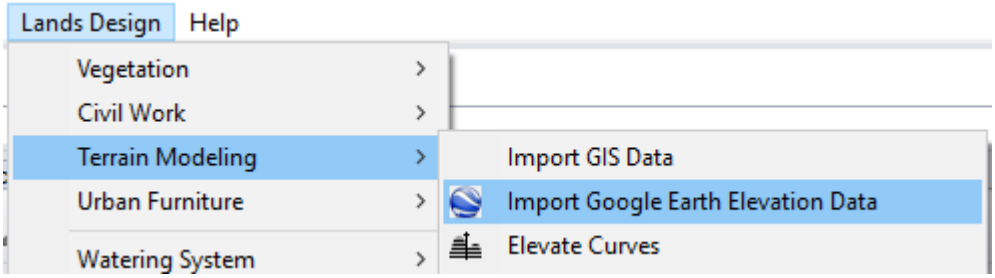
Nous allons nous attarder sur la création de la carte en trois dimensions et son importation dans Unity. Pour cela il nous a fallu faire quelques recherches sur les possibilités que nous avons pour créer cette carte sans devoir avoir de solides connaissances dans des outils de création de modèles 3D.

La façon la plus simple a été de récupérer depuis Google Earth un scan de la carte avec son relief et ensuite depuis un logiciel de modélisation 3D exporter la carte en modèle utilisable dans Unity.

Les logiciels utilisés sont Rhinocéros 5, pour la modélisation, avec le module d'extension Lands Design qui lui nous permet d'aller récupérer la carte sur Google Earth. Ces logiciels sont payant mais ont une licence d'essai de 60 jours. Ci-dessous la marche à suivre afin de récupérer la carte et l'importer dans Unity :

Une fois le logiciel Rhinoceros 5 et Lands Design lancé, il faut se rendre dans le menu « Lands Design », ensuite le sous-menu « Terrain Modeling » et pour finir « Import Google Earth Elevation Data ».

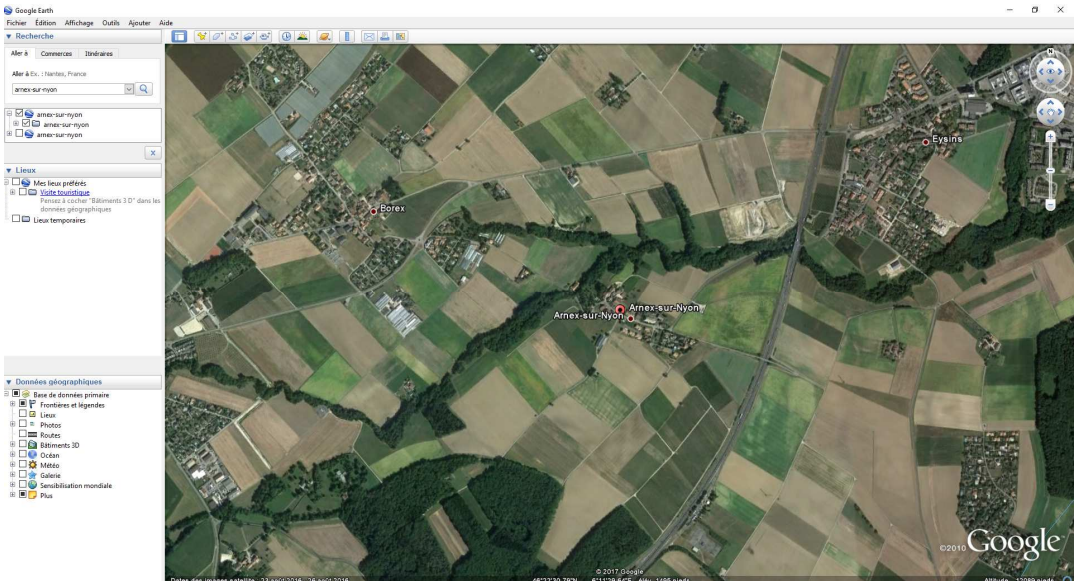
Figure 24: importer carte 3D #1



Source : personnelle

Une fois la fenêtre de Google Earth ouverte, il faut simplement se focaliser sur l'endroit voulu puis retourner dans Rhinocéros et cliquer sur « OK » en ayant au préalable vérifié que la coche « Google Earth image center » est activée.

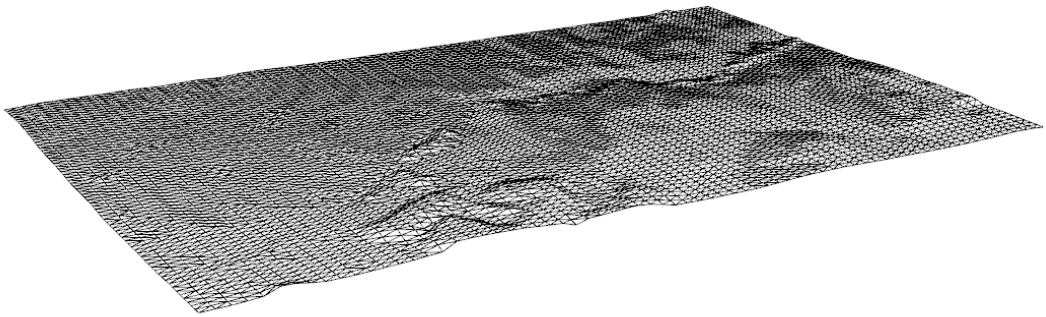
Figure 25: importer carte 3D #2



Source : personnelle image origine : Google Earth

Une fois l'opération terminée, nous pouvons voir le terrain modéliser avec l'élévation sur Rhinocéros.

Figure 26: importer carte 3D #3



Source : personnelle

Enregistrer ce modèle en format FBX, puis dans Unity dirigez-vous vers l'onglet « Assets », puis sélectionner « Import New Asset », allez chercher votre terrain. Afin d'appliquer une couleur au terrain il va falloir créer un nouveau matériel dans Unity et y assigner l'image tirée de Google Earth, puis appliquer ce matériel au terrain dans Unity.

9.2.1.4 Créer nos athlètes et les afficher sur la carte

Maintenant que nous avons nos objets ainsi que la carte en trois dimensions nous pouvons nous adonner à la création du visuel des athlètes sur la carte.

Pour cela nous allons créer dynamiquement, c'est-à-dire lorsque nous lançons l'application, des objets appelés « GameObject ». Ils auront le visuel que nous déciderons, pour le moment ce visuel se restreint à de simples rectangles dont la couleur est générée de manière aléatoire à leur création.

Pour créer ces « GameObject », nous devons connaître le nombre d'athlètes présent pour la course. Nous pouvons le savoir en comptant le nombre d'objet athlète dans la collection qui se trouve dans l'objet « Groups » qui lui-même fait partie de la collection présente dans l'objet « Timestamp ». Par exemple, avec le code suivant :

```
foreach (Athlete a in listGroups) { }
```

Nous pouvons effectuer une action spécifique comme créer un GameObject qui représentera le coureur sur la carte pour chaque athlète présent dans le groupe.

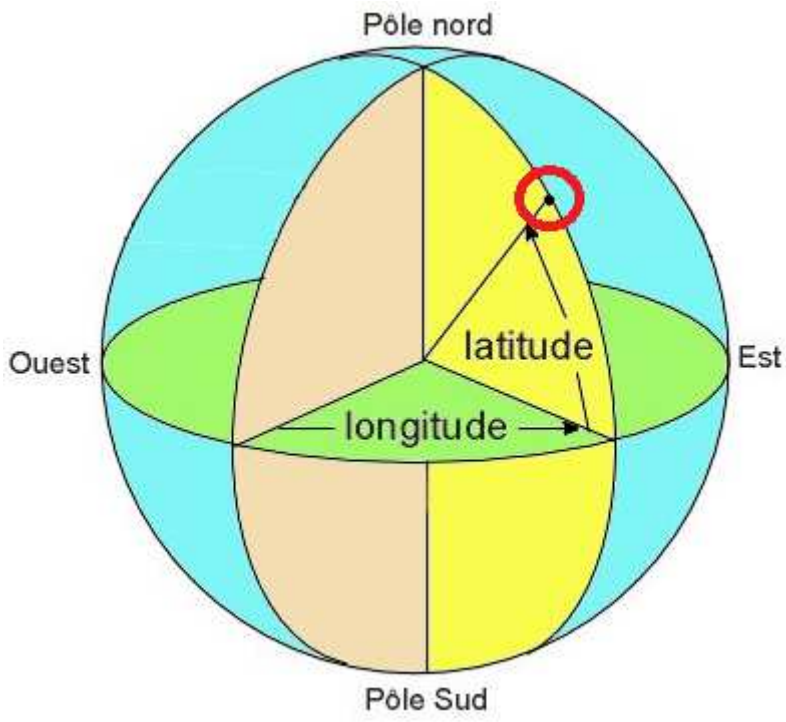
A présent, nos participants sont affichés sur la carte mais pas au bon endroit. Il nous faut donc connaître leur position, et ce point est détaillé dans la prochaine section.

9.2.1.5 Normalisation des données

Nous recevons de la part de GPST les données GPS pour chaque athlète, nous connaissons donc la latitude ainsi que la longitude des coureurs. Mais ces données correspondent au monde réel et non au monde de Unity.

Le schéma suivant nous aide à comprendre ce que sont les données GPS.

Figure 27: schéma coordonnées GPS



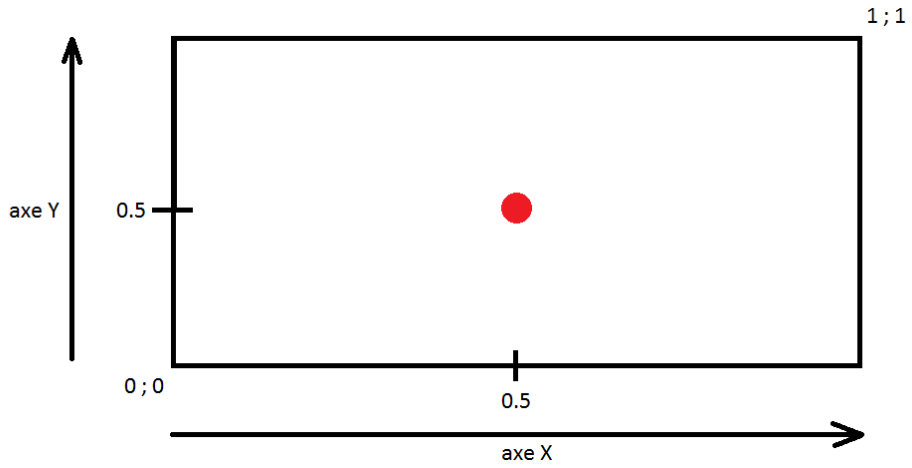
Source : image modifiée par l'auteur, source originale : https://zestedesavoir.com/tutoriels/857/lastronomie-depuis-mon-canape/ca-tourne-pas-droit/

Elles nous donnent donc un point situé sur une sphère. Mais comme dit plus haut ces coordonnées sont pour le monde réel, nous devons donc transformer ces coordonnées afin de correspondre au monde créé par Unity.

Nous allons donc normaliser nos données GPS. Les schémas suivant vont vous aider à mieux comprendre le principe de la normalisation.

Prenons par exemple un monde de taille minimum 0 et maximum 1. Nous avons notre point en coordonnées 0.5 ; 0.5. Soit x = 0.5 et y = 0.5.

Figure 28: schéma normalisation #1



Source : personnelle

Nous voulons recréer un monde qui cette fois sera de taille minimum -1 et maximum 1. Nous allons donc devoir normaliser les coordonnées de notre point afin de correspondre aux nouvelles dimensions de notre nouveau monde.

La formule mathématique pour effectuer cette transformation est la suivante :

$$x = \frac{x_{coordonnée} - x_{\min source}}{x_{\max source} - x_{\min source}}, \qquad y = \frac{y_{coordonnée} - y_{\min source}}{y_{\max source} - y_{\min source}}$$

Soit, en utilisant des vraies valeurs :

$$x = \frac{0.5 - 0}{1 - 0}, \qquad y = \frac{0.5 - 0}{1 - 0}$$

Nous allons ensuite effectuer l'opération suivante afin de trouver les coordonnées de notre nouveau point :

$$x' = x * (x_{\max destination} - x_{\min destination}) + x_{\min destination}$$

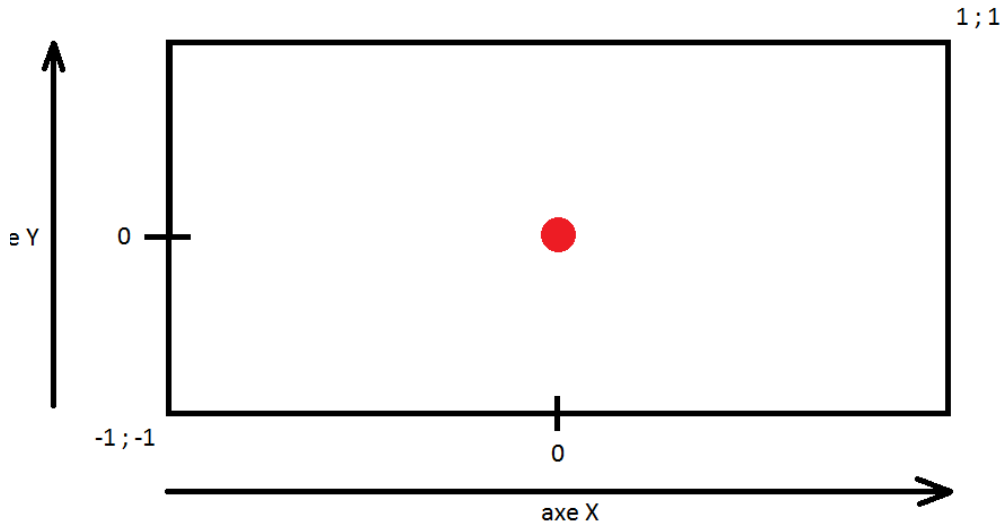
$$y' = y * (y_{\max destination} - y_{\min destination}) + y_{\min destination}$$

Soit avec nos vraies valeurs :

$$x' = 0.5 * \big(1 - (-1)\big) + (-1), \qquad y' = 0.5 * \big(1 - (-1)\big) + (-1)$$

Nos nouvelles valeurs seront donc pour l'axe des x : 0 et pour l'axe des y : 0.

Figure 29: schéma normalisation #2



Source : personnelle

En comparant les deux schémas nous pouvons voir que le point est toujours au centre malgré les nouvelles dimensions de notre nouveau monde.

Pour revenir à notre prototype nous avons donc des coordonnées GPS qui sont de 46.3604 pour la latitude et 6.205 pour la longitude. Premièrement nous devons connaître les valeurs minimum et maximum de notre monde :

Figure 30: carte Google Earth

Source : Google Earth

Le cercle jaune représente le point des coordonnées minimum de notre carte et le cercle vert le maximum, nous voulons donc la latitude et la longitude minimum et maximum de ces points.

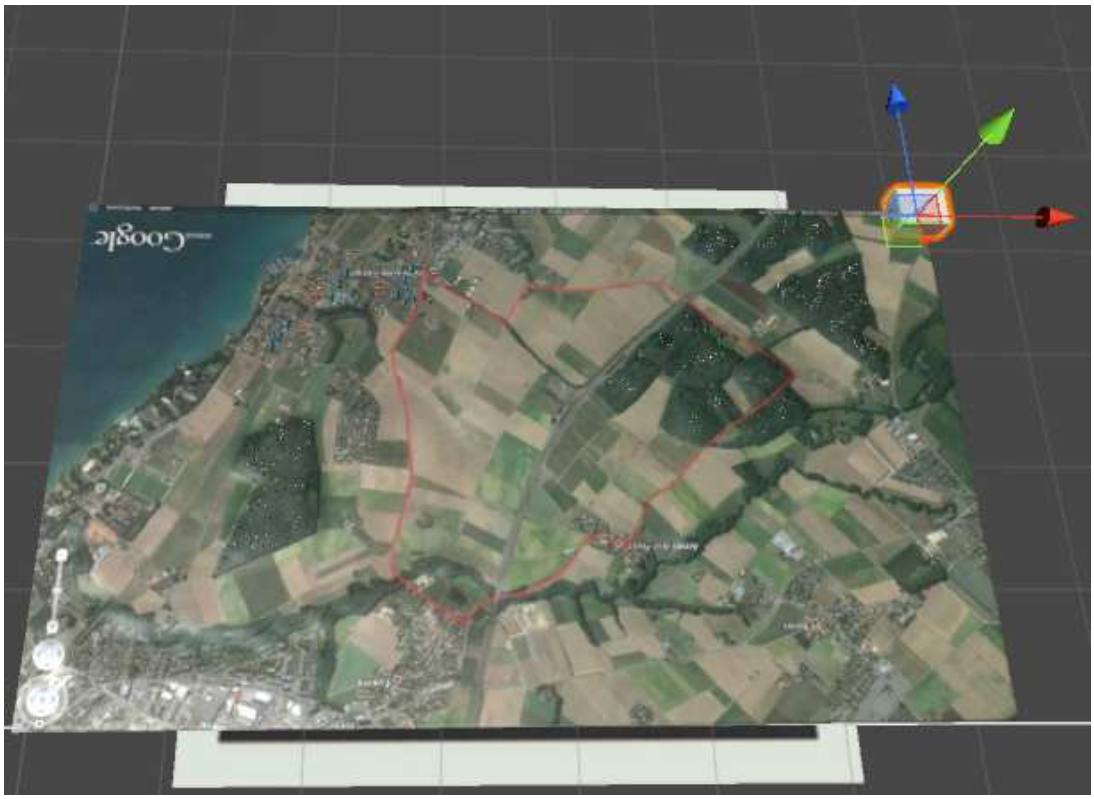
- Maximum latitude = 46.383687
- Minimum latitude = 46.355645
- Maximum longitude = 6.23532
- Minimum longitude = 6.165085

Dans notre monde virtuel de Unity nos coordonnées minimum et maximum vont totalement changer. Nous allons donc avoir comme valeurs :

- Maximum X = 1.414
- Minimum X = 0
- Maximum Y = 0.838
- Minimum Y = 0

Afin de découvrir ces valeurs dans Unity nous devons créer un container qui agira comme point d'origine, ensuite nous ferons du terrain un enfant de ce dernier. Pour connaître les coordonnées maximum, la manière la plus simple que nous ayons trouvé est de créer un simple cube puis de le placer au coin supérieur représentant les valeurs maximums, comme l'image ci-dessous :

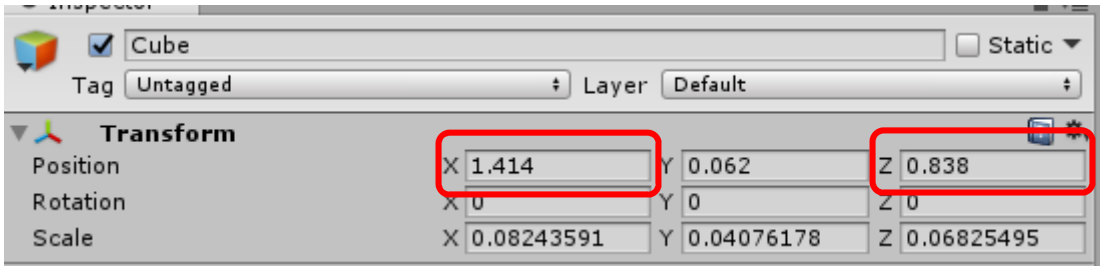
Figure 31: point maximum Unity



Source : personnelle

Ensuite nous pouvons observer le panneau qui indique la position de l'objet :

Figure 32: position Unity



Source : personnelle

La référence de l'axe X donne la position pour notre valeur maximum des X et l'axe Z nous référence sur la valeur maximum des Y.

Maintenant nous n'avons plus qu'à appliquer notre formule mathématique afin de normaliser nos données GPS et ainsi créer de nouvelles coordonnées qui seront fidèles à notre monde Unity tout en gardant leurs proportions de base. L'image suivante nous montre le code utilisé afin de normaliser les données :

Figure 33: code de normalisation

```
public Cartesian Normalized_Point(Athletes athlete)
{
    double x = (athlete.lat - min_Lat) / (max_Lat - min_Lat);
    double y = (athlete.lon - min_Lon) / (max_Lon - min_Lon);

    x = x * (MAX_X - MIN_X) + MIN_X;
    y = y * (MAX_Y - MIN_Y) + MIN_Y;

    return new Cartesian(athlete, x,y);
}
```

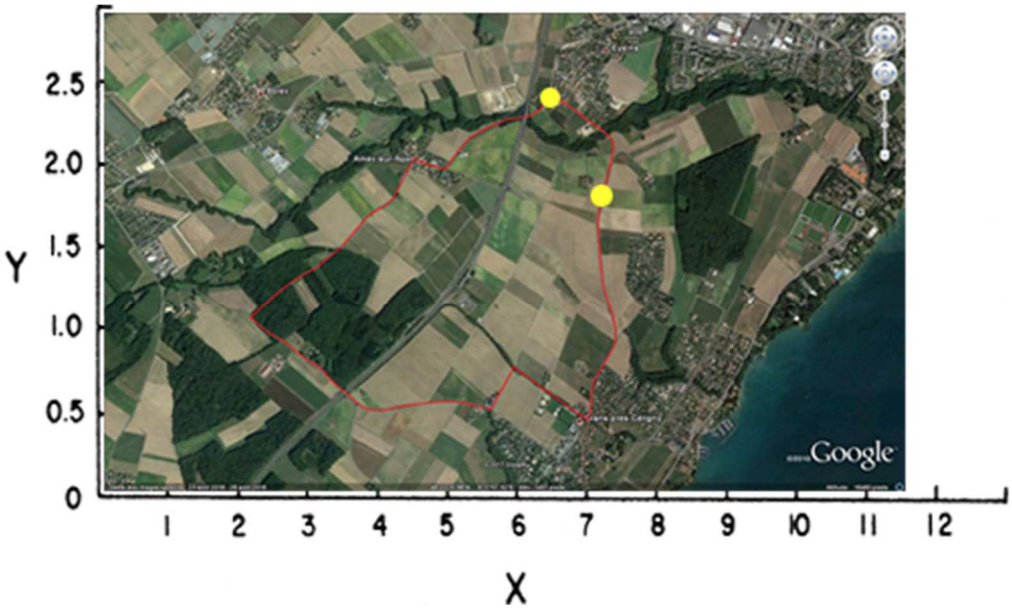
Source : personnelle

A ce stade nous avons donc des données utilisables au sein de Unity afin d'afficher un cycliste sur la carte au bon endroit. Mais en réalisant des tests nous allons nous apercevoir d'un problème, le point d'origine des calculs (Figure 30: carte Google Earth, cercle jaune) est différent entre la réalité et notre monde Unity. La résolution du problème sera expliquée au point suivant.

9.2.1.6 Normalisation, problème d'origine

Lorsque nous lançons notre programme, nous nous apercevons que les points ne sont pas générés à la bonne place. Après réflexion, nous nous rendons compte qu'en réalité les points sont positionnés au bon endroit mais avec une origine différente, explications :

Figure 34: normalisation origine humaine



Source : image modifiée. Sources : Google Earth, diagramme : http://www.fao.org/docrep/003/X6845F/X6845F02.htm

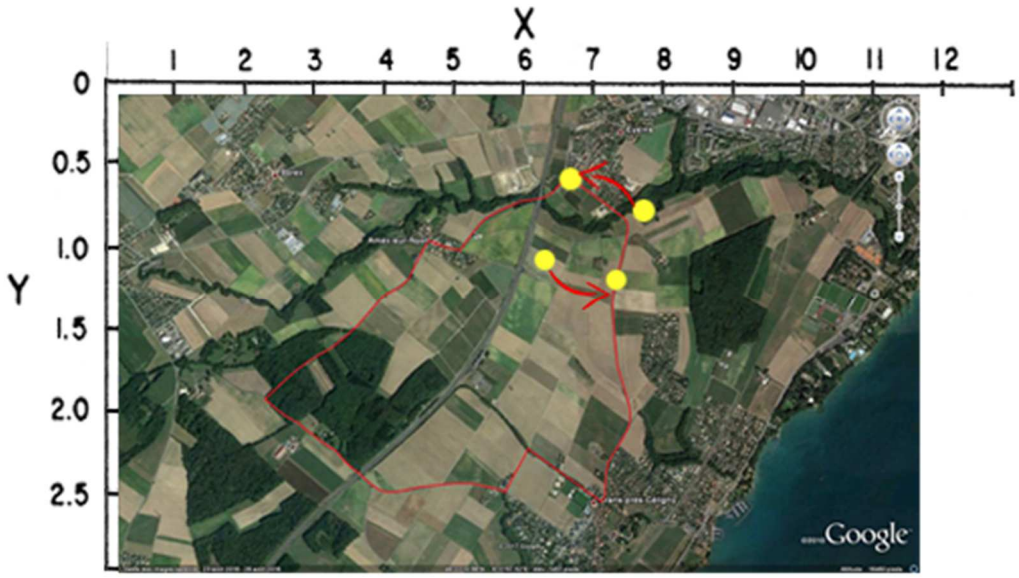
Ci-dessus en jaune nous avons les coordonnées de deux de nos athlètes. Nous pouvons observer que notre point d'origine est en bas à gauche. Nous définissons nous-même que l'origine doit être en bas à gauche car il est logique de s'imaginer un axe des Y grandissant vers le haut. Mais l'ordinateur, dans sa grande sagesse, définit l'origine en haut à gauche avec un axe Y grandissant vers le bas. Étant donné que l'origine est différente de celle que nous avons choisis, nous allons devoir effectuer une rotation des points et les étirer légèrement afin que ces derniers correspondent à l'origine définit par l'ordinateur.

Figure 35: normalisation origine ordinateur

Source : image modifiée. Sources : Google Earth, diagramme : http://www.fao.org/docrep/003/X6845F/X6845F02.htm

Ci-dessus nous avons une illustration de la vision de l'ordinateur (les points jaunes sont placés de manière à illustrer et ne sont pas représentatif du vrai placement fait par l'ordinateur). Comme dit plus haut nous allons devoir procéder à une rotation ainsi qu'un étirement des points afin qu'ils correspondent à l'origine de l'ordinateur.

Figure 36: rotation et étirement des points



Source : image modifiée. Sources : Google Earth, diagramme : http://www.fao.org/docrep/003/X6845F/X6845F02.htm

Ci-dessous la formule qui nous permet de réaliser cette rotation et étirement :

Figure 37: code de rotation et étirement

```
public Cartesian convertPoints(Cartesian cartesian)
{
    double x1 = -(cartesian.y / (MAX_Y - MIN_Y)) * (MAX_X - MIN_X) + MAX_X;
    double y1 = -(cartesian.x / (MAX_X - MIN_X)) * (MAX_Y - MIN_Y) + MAX_Y;

    return new Cartesian(cartesian.athlete, x1,y1);
}
```

Source : personnelle

x1 et y1 sont donc nos points correctement placés sur la carte de Unity.

9.2.1.7 Définir la position des objets représentant les cyclistes sur la carte en 3D

À cet instant nous avons tous les éléments afin de créer et positionner correctement nos cyclistes dans notre monde en trois dimensions. Nos coureurs étant déjà créés et affichés sur la carte il nous reste plus qu'à définir leur position. Pour réaliser cette action nous allons faire appel aux fonctions de Unity, en implémentant à nos scripts la classe de base MonoBehaviour nous pouvons accéder à des méthodes interne à Unity afin de déterminer la position d'un GameObject. Le code suivant nous montre comment créer un objet de type GameObject en lui donnant comme aspect ce que nous avons défini afin de représenter un cycliste, de déterminer sa position ainsi que sa rotation et de le rendre enfant d'un autre GameObject qui aura comme fonction d'être un espace réservé afin de contenir les objets créés.

Figure 38: code de génération des cyclistes

```
GameObject go = Instantiate(cyclistPrefab,
new Vector3((float)c.x, 0, (float)c.y),
Quaternion.identity, point0.transform) as GameObject;
```

Source : personnelle

c.x et c.y représente la position sur les axes et en créant un nouveau Vector3 nous pouvons déterminer la position de l'objet dans le monde de Unity.

Les objets représentant les athlètes sont créés une seule et unique fois, lors de la réception du premier fichier contenant les données en format JSON. Ensuite nous ne faisons que bouger ces objets sur le terrain en redéfinissant leur position pour chacun.

9.3 Architecture des dossiers du prototype dans Unity

Unity crée une arborescence de dossiers avec comme racine un dossier nommé « Assets ». Sous ce dossier nous allons pouvoir créer une multitude de sous dossiers ayant chacun sa spécificité.

Les dossiers suivants font partie de dossiers regroupant soit des modèles 3D téléchargés depuis l'Asset Store de Unity (site internet nous permettant de télécharger gratuitement ou en payant des modèles 2D ou 3D, des scripts, des systèmes de particules, etc… créés par la communauté), soit des scripts de base de Vuforia :

- Area 730

  Ce dossier provient de l'Asset Store, il regroupe plusieurs modèles de maisons et bâtiments en trois dimensions, ils servent à la démonstration d'objets 3D sur la carte.
- Common

  Ce dossier regroupe des scripts de Vuforia.
- Elemental Free

  Toujours en provenance de l'Asset Store, une multitude de systèmes de particules sont présent dans ce dossier.
- Olive_tree

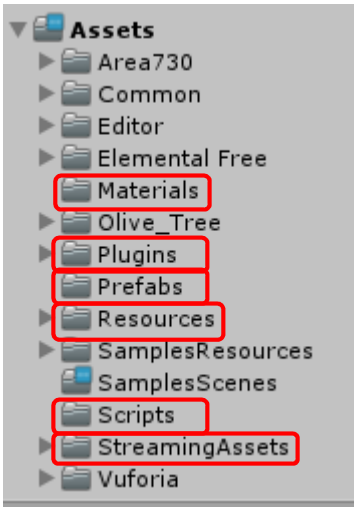
  Tout comme Area 730, ce dossier contient un arbre et a permis de créer les forêts en trois dimensions.
- Samples Ressources

  Dossier de Vuforia qui regroupe scripts et matériaux.
- Samples Scenes

  Des exemples d'utilisation des outils de Vuforia sous forme de scènes.
- Vuforia

  Dossier d'importation du Framework de Vuforia, il contient tout ce dont on a besoin afin de développer en réalité augmentée.

Figure 39: architecture des dossiers Unity



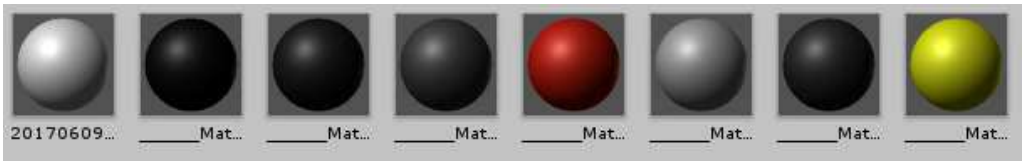
Source : 1personnelle

Les dossiers dans les rectangles rouge sont ceux qui nous intéressent le plus, et également ceux créés par nous. Nous avons donc les dossiers suivants.

9.3.1 Dossier Materials

Il regroupe toutes les textures et couleurs que nous avons utilisé dans notre application.

Figure 40: dossier Materials



Source : personnelle

Source : 2personnelle

9.3.2 Dossier Plugins

Ce dossier nous l'avons créé au début du projet et nous y avons ajouté la librairie LitJson afin que nous puissions l'utiliser dans les scripts du prototype. Lorsque nous lancerons la production d'une application Android (APK) le dossier Android se créera automatiquement dans le dossier Plugins et y ajoutera tous les éléments nécessaires au bon fonctionnement de l'application sous Android. Il en va de même pour iOS.

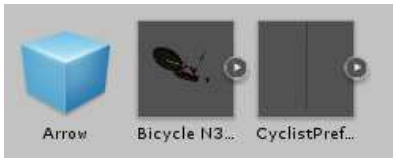
Figure 41: dossier Plugins

Source : personnelle

9.3.3 Dossier Prefabs

Il contient tous les éléments que nous avons conçu dans Unity et qui sont utilisés en tant que GameObject. Nous pouvons retrouver la « prefab » des cyclistes qui est pour l'instant un simple rectangle ainsi que le modèle de la flèche qui indique la position sur la carte des athlètes lorsque notre écran ne les a plus dans son focus.

Figure 42: dossier Prefabs

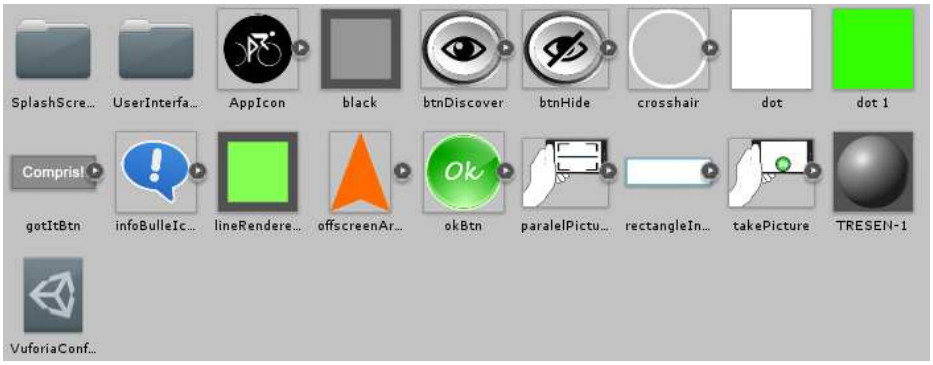


Source : personnelle

9.3.4 Dossier Ressources

Le dossier des ressources correspond aux images utilisées dans notre application. Par exemple, nous créons un bouton et il a comme image de base un rectangle blanc. Nous voulons que ce bouton soit rond et ressemble à un œil en son milieu, nous allons donc créer une image et la placer dans ce dossier. Une fois l'image élaborée et importée dans Unity il nous suffit de la sélectionner et de la placer sur le bouton, le rectangle blanc par défaut sera remplacé automatiquement.

Figure 43: dossier Ressources

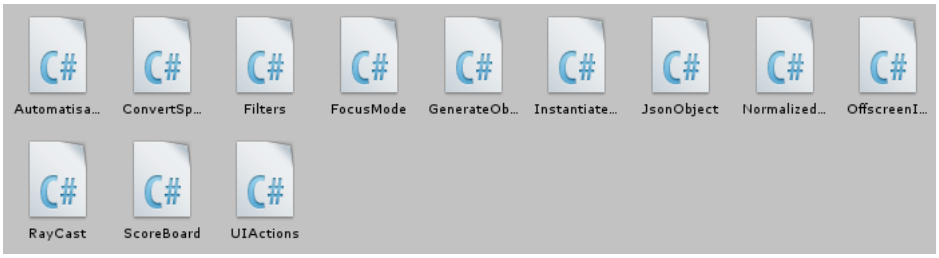


Source : personnelle

9.3.5 Dossier Scripts

Comme son nom l'indique ce dossier contiendra tous nos scripts, il est possible de réaliser des sous-dossiers afin de mieux séparer chaque script si nous avons plusieurs scènes par exemple.

Figure 44: dossier Scripts



Source : personnelle

9.3.6 Dossier Streaming Assets

Ce dossier va nous permettre de récupérer des dossiers qui se situeront directement dans l'APK généré par Android. Il nous est donc possible d'y placer des fichiers et de les retrouver une fois l'application établie pour être utilisée sur notre smartphone, dans notre contexte se sont des fichiers contenant les données en format JSON. En plus des fichiers que nous y avons placés, Vuforia y ajoute également quelques fichiers pour sa propre configuration. Notre dossier contenant les fichiers JSON ne sera plus nécessaire dès le moment où l'application utilisera des données mis en ligne par GPST.

Figure 45: dossier StreamingAssets #1

Source : personnelle

Figure 46: dossier StreamingAssets #2

Source : personnelle

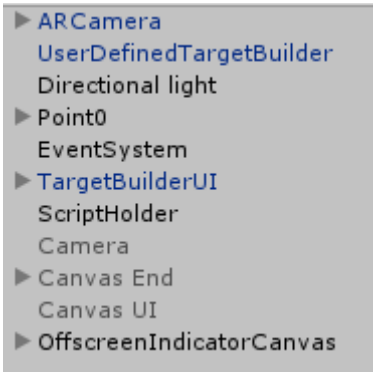
Pour l'instant notre prototype utilise des données locales afin de simuler une course cycliste. Au terme de son développement elle devra être capable de recourir à des données provenant d'un web-service et fournit par l'entreprise GPST. Bien entendu, ce prototype est codé de façon à ce que l'implémentation de cette technologie se fasse le plus simplement possible à l'avenir, peu d'adaptation devront être fourni.

9.4 Architecture du prototype dans Unity Editor

Dans cette section nous allons voir les éléments présents dans notre scène, une scène définie l'univers actuelle dans lequel nous allons placer notre environnement, nos obstacles et décorations.

Notre projet se présente comme tel dans Unity Editor :

Figure 47: architecture Unity Editor

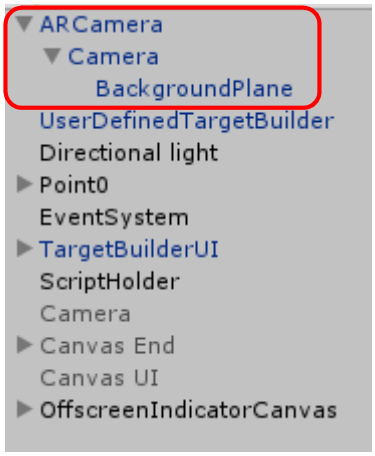


Source : personnelle

Afin d'avoir une meilleure vue nous allons segmenter chaque dossier et les observer individuellement.

Nous commençons par le premier dossier, « ARCamera ».

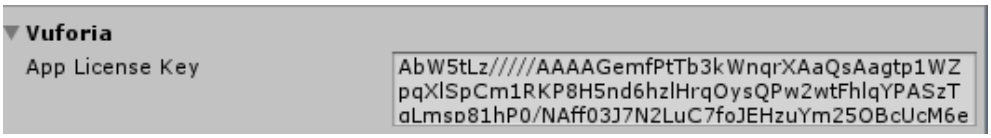
Figure 48: objet ARCamera



Source : personnelle

Il nous permet l'utilisation de la caméra de notre appareil afin d'y détecter le marqueur. Lorsque nous cliquons sur l'objet ARCamera nous pouvons accéder aux configurations de celle-ci, et ainsi y placer la licence créée au préalable sur le portail développeur de Vuforia.

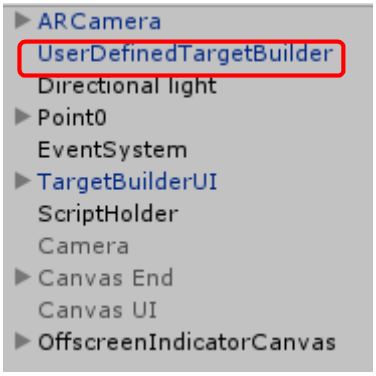
Figure 49: ARCamera configuration



Source : personnelle

L'objet suivant, « UserDefinedTargetBuilder », est un espace qui nous sert à définir les propriétés pour la technologie du User Define Target (UDT), elle permet à l'utilisateur de définir son propre marqueur afin de bénéficier de la réalité augmentée.

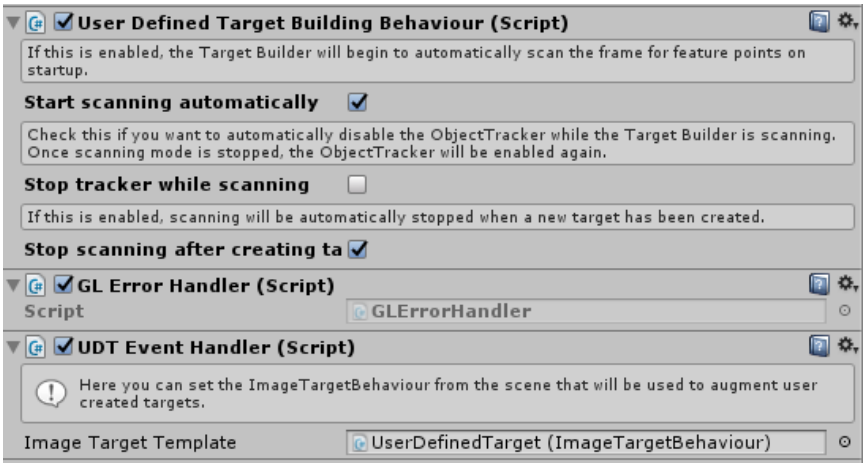
Figure 50: objet UserDefinedTargetBuilder



Source : personnelle

Ci-dessous nous pouvons observer les scripts propres à Vuforia rattachés à cet objet :

Figure 51: configuration UserDefinedTargetBuilder

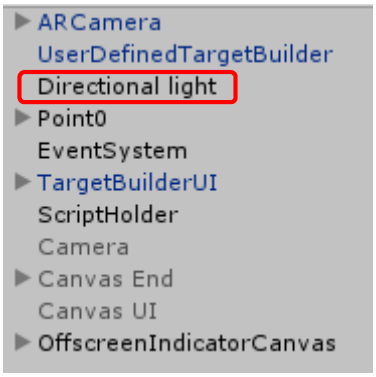


Source: 3personnelle

- « Start scanning automatically » lance, comme son nom l'indique, le scan automatique de l'environnement vu par la caméra.

- « Stop tracker while scanning » désactive le ObjectTracker (la reconnaissance d'image/objet) pendant la période de scan.
- « Stop scanning after creating target » permet de stopper le scan de l'environnement après la création d'un marqueur.

Figure 52: objet Directional Light

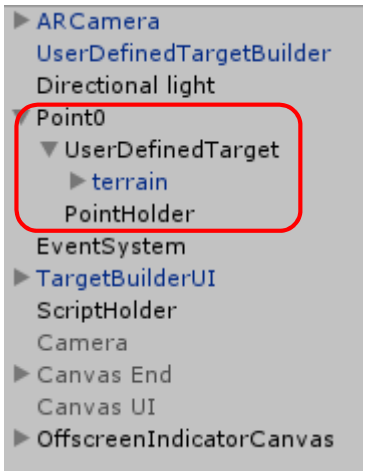


Source : personnelle

« Directional Light » est une simulation du soleil sur notre environnement, sans cette lumière tout serait plongé dans une pénombre comme au crépuscule.

L'objet suivant est notre point d'origine pour notre carte en relief et contient également l'objet parent de nos futures GameObject créent dynamiquement qui représenteront nos cyclistes.

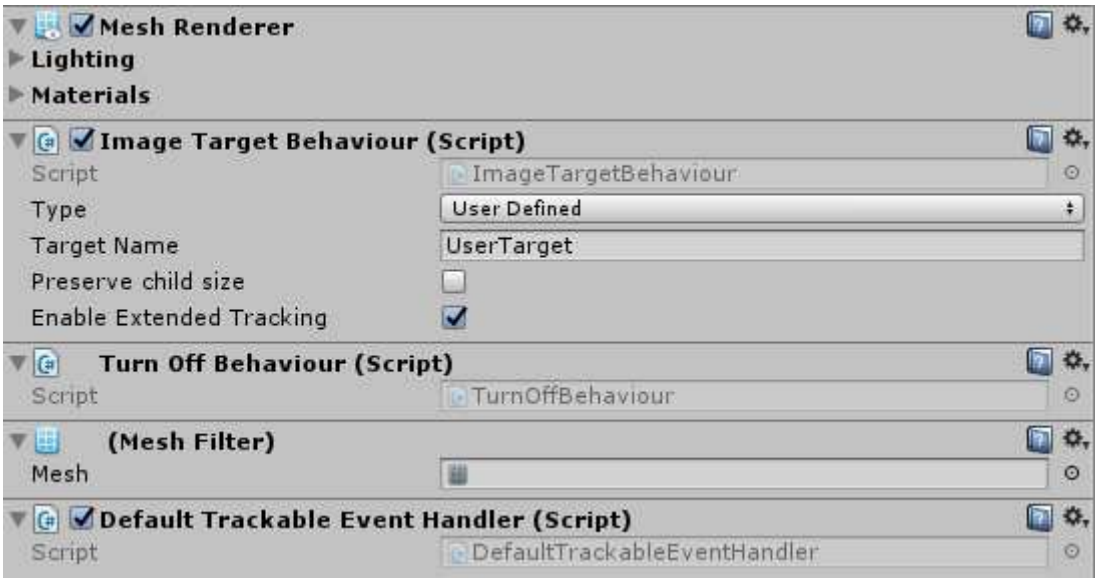
Figure 53: objet Point0



Source : personnelle

Le premier objet enfant de « Point0 » est « UserDefineTarget », cet objet possède les scripts de référence de Vuforia :

Figure 54: objet UserDefineTarget configuration



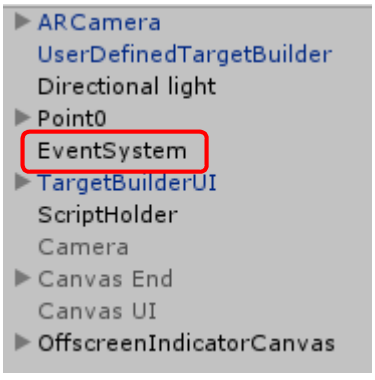
Source : 4personnelle

L'enfant de ce dernier sera notre terrain 3D. Dès que le marqueur est détecté le terrain va être placé dans son univers et permettre ainsi sa visualisation au travers de la caméra pour l'utilisateur.

L'objet « PointHolder », qui est également un enfant de « Point0 », est l'objet parent des GameObject représentant les athlètes.

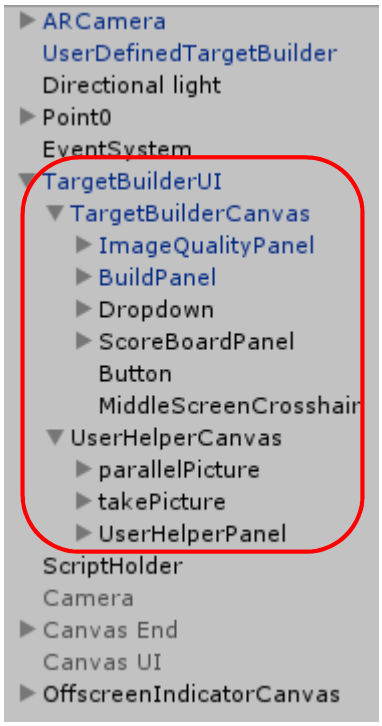
L'objet « EventSystem » est créé automatiquement par Unity lors de la première exécution du programme, il permet de connaître les commandes entrantes faite par l'utilisateur comme une pression sur une touche du clavier ou un toucher sur l'écran tactile.

Figure 55: objet EventSystem



Source : 5personnelle

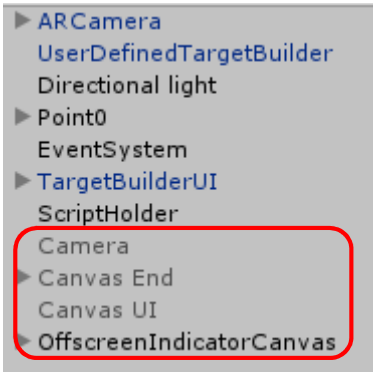
Figure 56: objet TargetBuilderUI



Source : personnelle

« TargetBuilderUI » regroupe les éléments graphiques que l'utilisateur voit sur son écran. Les informations quand l'image prise n'a pas la qualité suffisante et la petite aide afin de comprendre comment la prise d'image pour que le marqueur fonctionne sont affichées ou masquées selon les actions de ce dernier. Ces objets regroupent donc les éléments visuels de l'écran qui ne sont pas en réalité augmentée. Le script gérant ces éléments se trouve dans l'objet parent « TargetBuilderUI ».

Figure 57: objet Camera, CanvasEnd, CanvasUI, OffscreenIndicatorCanvas



Source : personnelle

L'objet « Camera » prend le relai quand la caméra de notre téléphone est coupée afin d'éviter un écran noir. Lorsque celle-ci est activée c'est qu'il n'y a plus de fichier JSON entrant et le visuel « Canvas End » apparaît.

« Canvas UI » est référencé dans un script mais n'est plus utilisé. Afin d'éviter tout erreur lors de l'exécution du programme il a été gardé. Il était utilisé lors de test initial quand la carte 3D n'était pas encore implémentée dans le projet.

« OffscreenIndicatorCanvas » permet de placer sur l'écran les flèches d'indication de la position des cyclistes lorsque nous ne les avons pas dans notre champ de vision.

Nous allons maintenant nous attarder sur un des objets les plus importants de notre scène, le « ScriptHolder ».

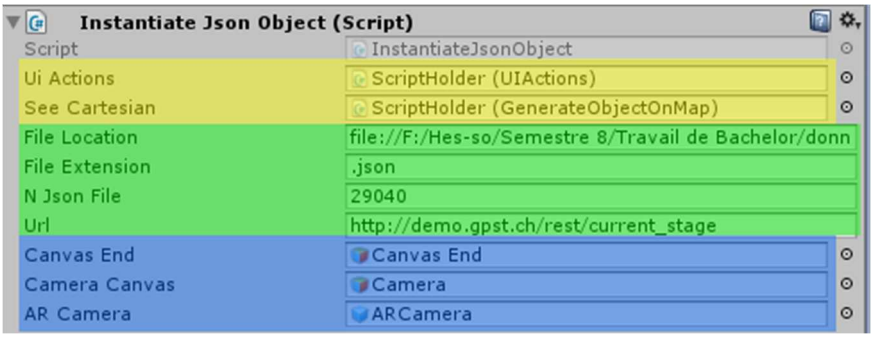
Figure 58: objet ScriptHolder

Source : personnelle

Cet objet contient tous nos scripts, de toute la mécanique de notre application (excepté les mécaniques du UDT). Si un changement de référence doit être apporté sur un script c'est dans les configurations du « ScriptHolder » qu'il faudra le faire.

Nous allons prendre chaque script attaché à cet objet et en regarder les propriétés.

Figure 59: InstantiateJsonObject script



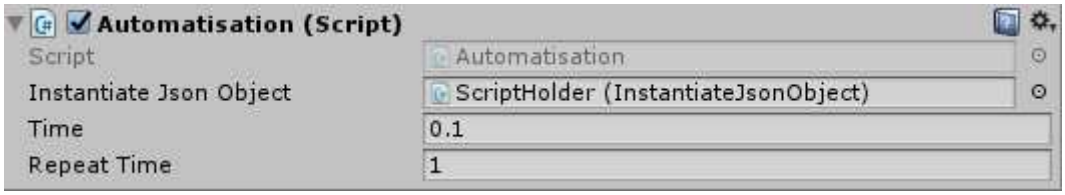
Source : personnelle

Dans l'encadré jaune nous avons les références aux autres scripts. Comme ils se trouvent tous dans le même objet container il suffit d'effectuer un glisser-déposer de l'objet dans le champ correspondant.

L'encadré vert contient les champs de références à l'emplacement des fichiers JSON, utiles uniquement lorsque nous testons l'application directement depuis le Unity Editor. Lors de l'utilisation sur un smartphone les fichiers se trouvent dans le Dossier Streaming Assets.

L'encadrer bleu fait référence à des éléments présent dans l'architecture de Unity Editor.

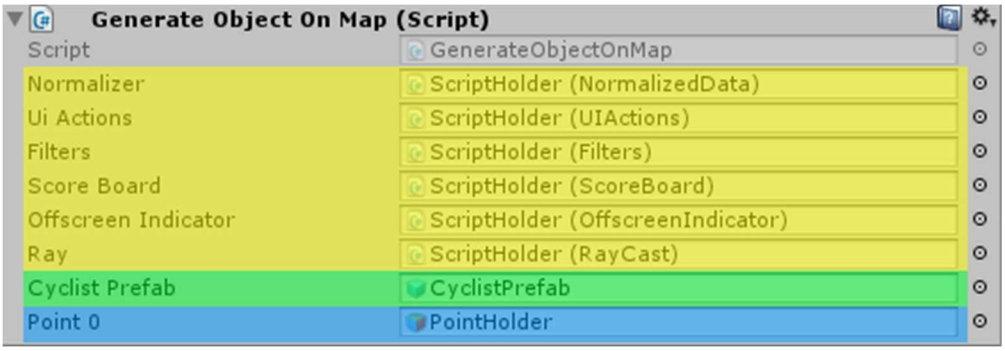
Figure 60: Automatisation script



Source : personnelle

Ce script automatise l'accès au fichier JSON. « Time » correspond à l'appel du premier fichier, un temps très cours est défini tandis que « Repeat Time » fait référence au temps entre chaque appel de fichier, ici le temps dit à notre application de charger un nouveau fichier JSON toutes les secondes.

Figure 61: GenerateObjectOnMap script



Source : personnelle

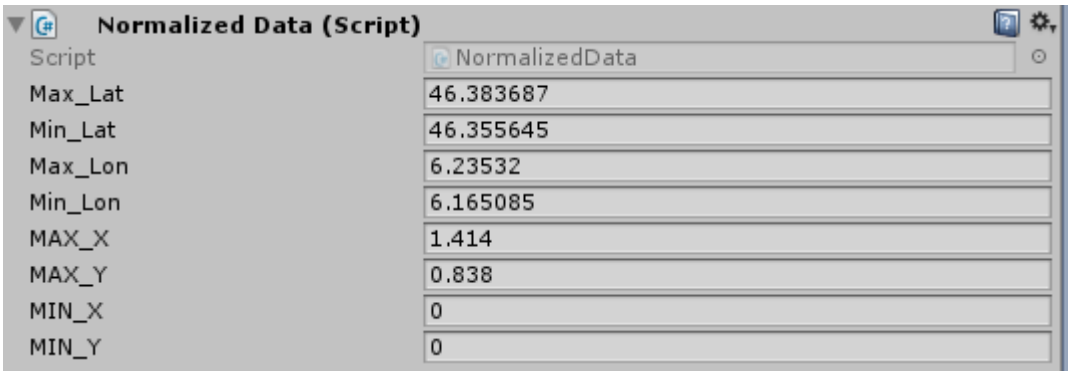
Dans l'encadré jaune nous retrouvons les références aux autres scripts toujours présents dans le « ScriptHolder ».

En vert c'est l'emplacement du modèle que nous allons utiliser pour représenter les cyclistes sur la carte. Si un nouveau visuel veut être employé il faudra le placer ici. Attention cependant, si le modèle choisit contient des éléments avec plusieurs couleurs il faudra très certainement faire attention avec le code suivant qui lui définit une couleur pour chaque GameObject de façon aléatoire :

```
go.GetComponent<Renderer>().material.color = new Color(Random.value, Random.value, Random.value);
```

En bleu c'est l'objet parent des GameObject créés dynamiquement.

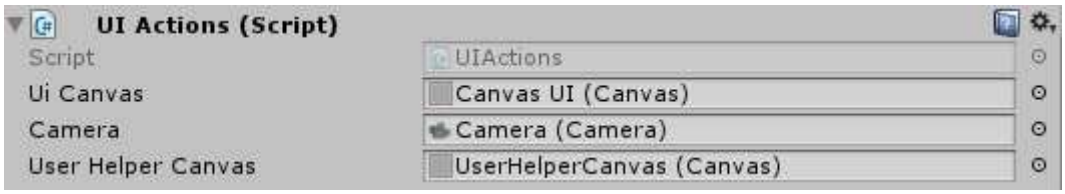
Figure 62: NormalizedData script



Source : personnelle

C'est dans ces champs qu'il faudra changer les valeurs afin de calibrer au mieux la transformation des coordonnées GPS en coordonnées interprétable pour Unity.

Figure 63: UI Actions script



Source : personnelle

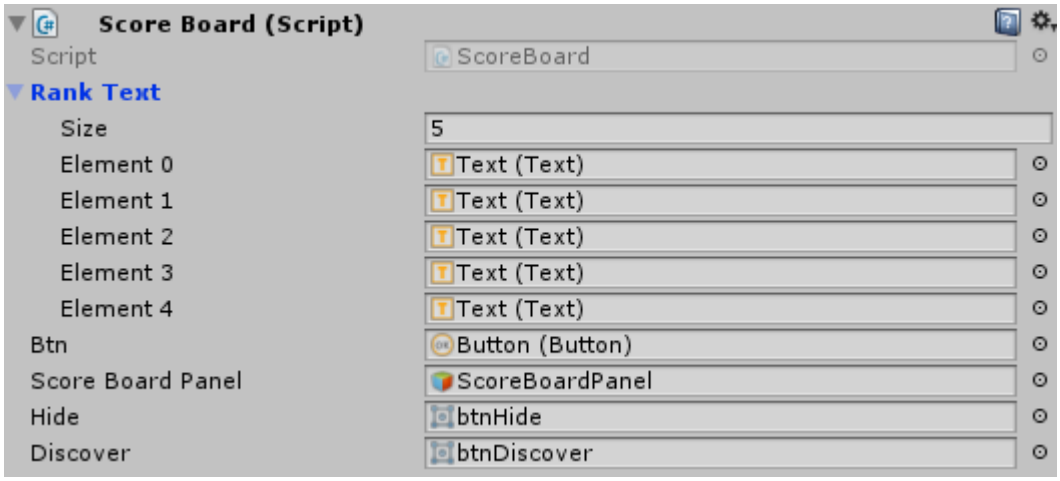
Script opérant des actions sur des éléments graphiques comme l'affichage à des moments donnés ainsi que leur désactivation. Principalement appliqué lors des premiers tests sans la carte, il reste néanmoins utilisé.

Figure 64: Filters script

Source : personnelle

Ce script gère la liste déroulante qui permet d'affecter un filtre sur les cyclistes.

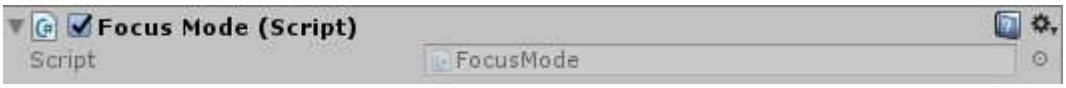
Figure 65: ScoreBoard script



Source : personnelle

La collection « Rank Text » regroupe les textes qui afficherons les cinq premiers de la course. « Btn » correspond au bouton qui affiche ou masque le tableau des résultats. « Hide » et « Discover » sont les images qui feront office de fond pour le bouton.

Figure 66: FocusMode script



Source : personnelle

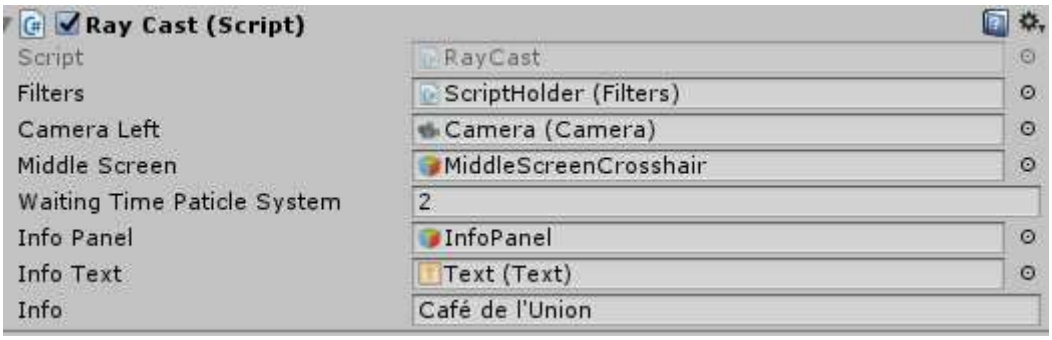
Le script « Focus Mode » permet à la caméra d'établir le focus de manière automatique.

Figure 67: Offscreen Indicator script

Source : personnelle

Ce script affiche des flèches indicatrices de la position des cyclistes sur la carte lorsqu'ils n'apparaissent plus sur l'écran. Les flèches se désactivent toutes seules dès lors que les GameObject représentant les athlètes sont de nouveau dans notre focus.

Figure 68: RayCast script

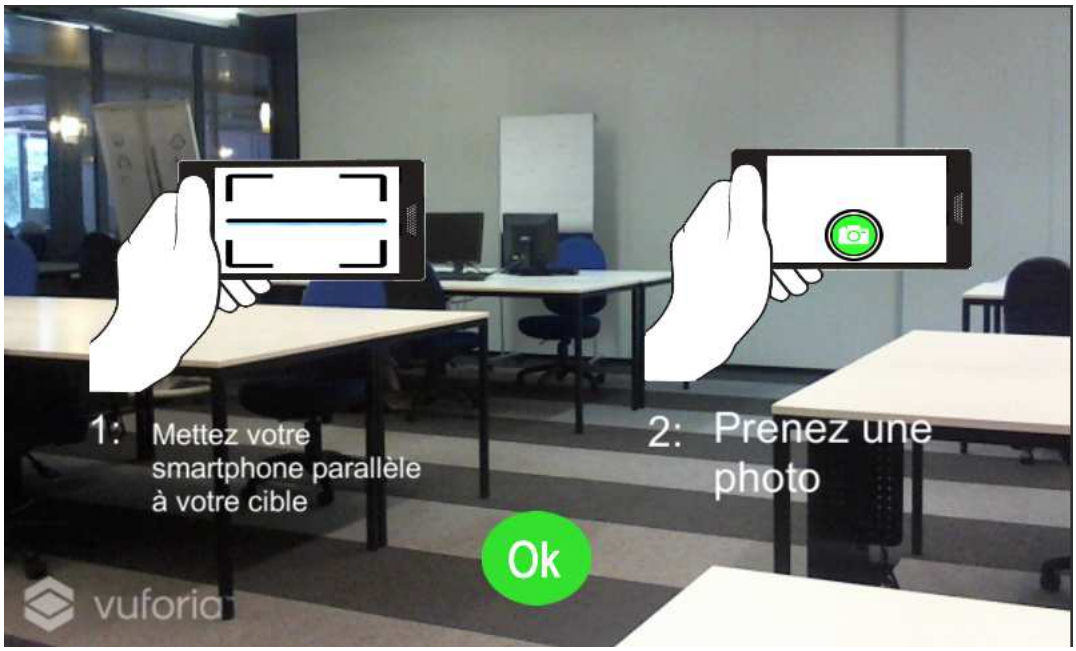


Source : personnelle

Ce script génère un rayon du centre de l'écran et détecte tous les objets avec lesquels il rentre en contact. Il nous permet de rendre en évidence des éléments sur notre carte 3D et nous donne également la possibilité d'ajouter des interactions sur notre univers en réalité augmentée sans utiliser des boutons à peser avec les doigts sur notre écran.

9.5 Images de l'application pendant son utilisation

Figure 69: image de l'application en utilisation #1



Source : personnelle

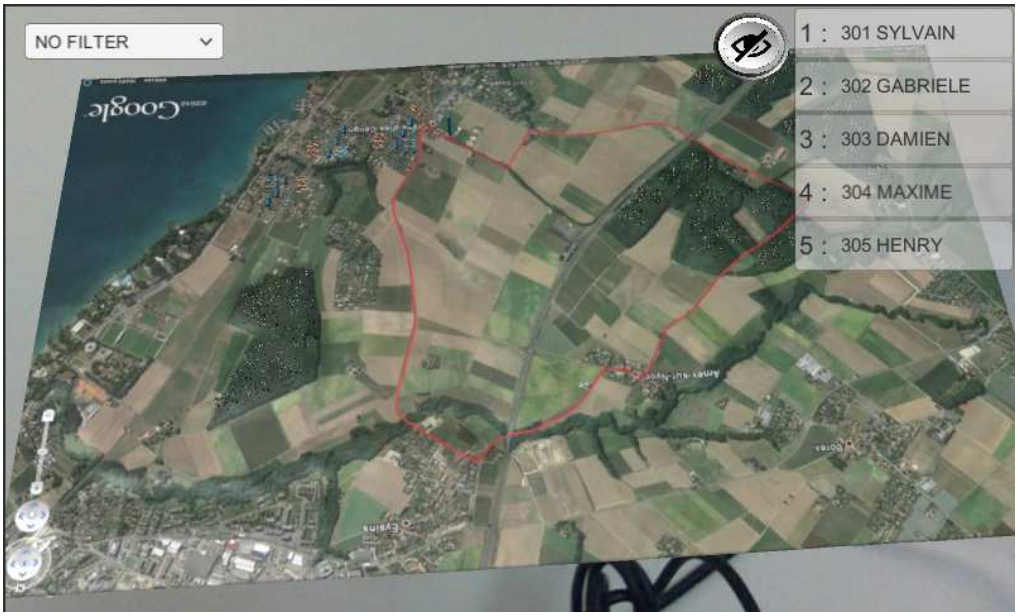
Voici la première image que nous pouvons observer lorsque nous lançant notre application. Un visuel nous aide à comprendre comment fonctionne la prise de vue afin de définir un marqueur.

Figure 70: image de l'application en utilisation #2

Source : personnelle

Si l'image ne contient pas suffisamment de détails ou est de mauvaise qualité, un message nous avertit qu'il faut refaire une photo.

Figure 71: image de l'application en utilisation #3



Source : personnelle

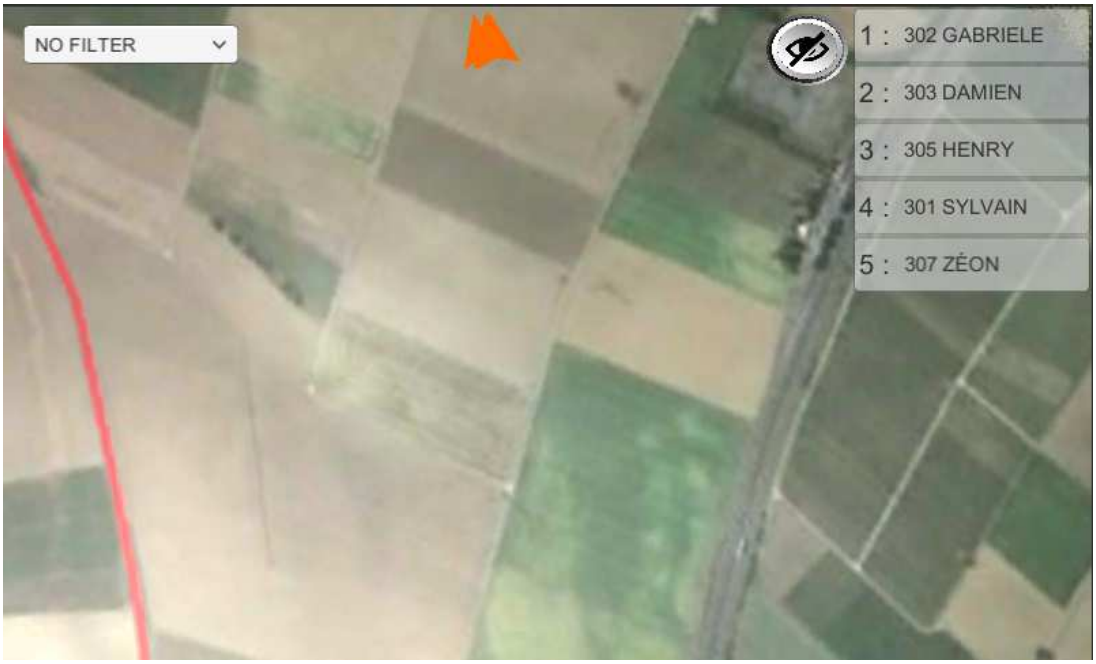
Lorsque l'image est bonne. La carte apparaît et la course commence, la liste déroulante avec les filtres possibles se définie ainsi que le rang des cyclistes dans le tableau des résultats.

Figure 72: image de l'application en utilisation #4

Source : personnelle

Nous distinguons les rectangles représentant les cyclistes et se mouvant le long du tracé.

Figure 73: image de l'application en utilisation #5



Source : personnelle

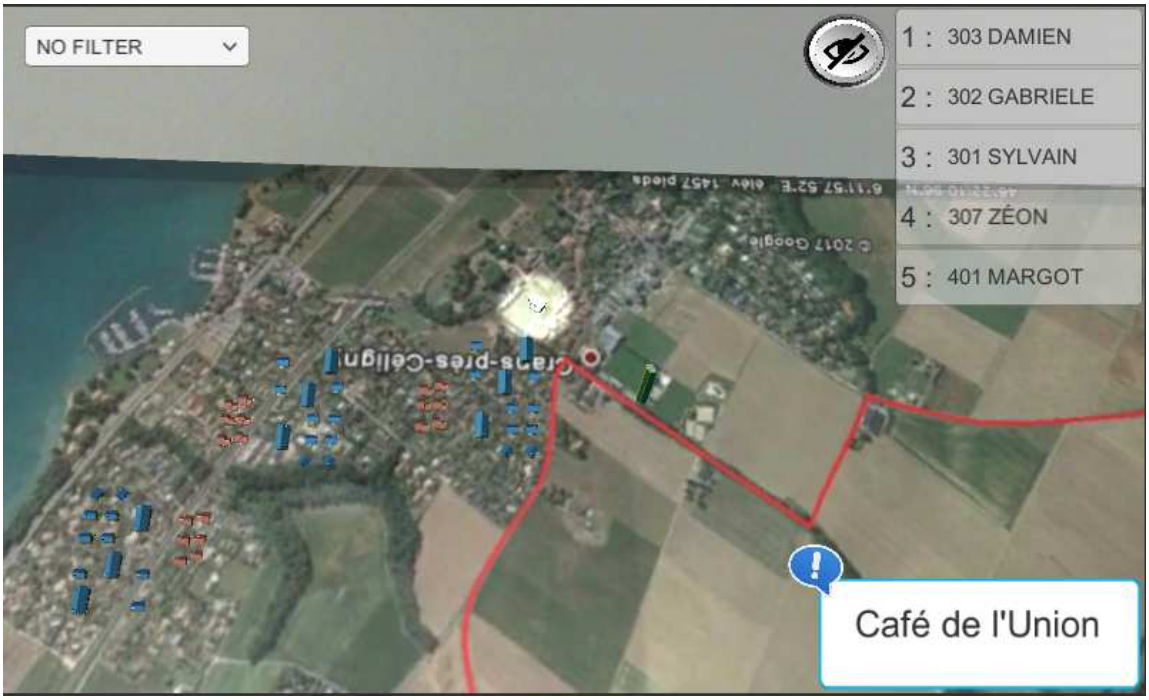
Les flèches indicatrices de la position des cyclistes lorsque ces derniers ne sont pas dans notre champ de vision.

Figure 74: image de l'application en utilisation #6

Source : personnelle

Ici nous avons caché le tableau des résultats et défini comme filtre de voir uniquement le participant avec le numéro de dossard 301.

Figure 75: image de l'application en utilisation #7



Source : personnelle

Nous avons un exemple de la mise en évidence d'un point d'intérêt. Il se déclenche automatiquement quand nous passons notre champ de vision par-dessus, il nous donne également l'information de ce que l'on voit.

10 Améliorations du prototype

La mécanique permettant l'utilisation de l'application comme vous avez pu la découvrir au chapitre précédent ne nécessite pas de grosse amélioration. Une relecture du code ainsi que l'ajout de tests afin d'éviter des erreurs lors du processus d'exécution sont peut-être nécessaires.

Les grosses améliorations à faire sur ce prototype concernent surtout les parties graphiques visible par l'utilisateur. Nous retrouvons donc :

- La liste déroulante des filtres

  En plus du fait que cette liste n'a pas de visuel personnalisé, il faudrait rajouter d'autres possibilités de filtre. Pour l'instant nous pouvons seulement filtrer les coureurs pour tous les afficher ou n'en afficher qu'un seul. Des filtres comme montrer uniquement le groupe, l'équipe, la nationalité sont tous possible par exemple. Nous pouvons reprendre tous les aspects de nos classes objets « Groups » et « Athletes » pour créer un nouveau filtre. Ces derniers ne concernent pas uniquement les cyclistes mais peuvent aussi affecter l'environnement, c'est-à-dire que l'utilisateur aurait la possibilité d'afficher ou non des éléments de la carte. En plus de revoir ses fonctionnalités une refonte graphique et peut-être un placement différent devront être effectué.
- Le tableau des scores

  À l'heure actuelle le tableau des scores affiche le numéro de dossard ainsi que le prénom de la personne. Ajouter des informations supplémentaires comme la nationalité peut apporter un plus. Des animations lors des changement de rang au sein des cinq premiers peuvent contribuer à un meilleur visuel.
- Le bouton d'activation et de désactivation du tableau des scores

  Revoir sa position ainsi que son graphisme. Mécaniquement parlant il fonctionne parfaitement et rempli son rôle mais sa position sur l'écran doit être revue.
- Les flèches indicatrices

  Les flèches sont l'un des éléments qui peuvent même être laissé tel quel, leur couleur orange assurant un bon visuel. Leurs formes et tailles peuvent cependant être améliorées.

- Les cyclistes

  L'élément représentant les cyclistes doit obligatoirement être changé. À titre d'essai pour le prototype les rectangles conviennent mais pour une application disponible au grand public une adaptation visuel doit être faite.

- Mise en évidence d'éléments sur la carte

  Cette mise en évidence vous pouvez l'observer à la Figure 75. Elle pointe sur un commerce existant, à l'avenir plus d'éléments de ce type auront la capacité d'être mis en évidence. Du commerce privé aux bâtiments public mais également des lieux naturels comme les forêts ou les lacs devront y figurer.

- La carte 3D

  Ce dernier point visuel n'est pas le moindre, au terme de ce prototype la carte possède bel et bien un relief comme vous pouvez le voir sur la Figure 26, mais l'image qui reproduit la carte est en deux dimensions et casse, en quelque sorte, l'effet de perspective. Afin d'y remédier, il faut faire appel à un designer 3D pour que nous puissions ajouter du contenu en trois dimensions sur la carte. Notre prototype contient quelques exemples de maisons et forêts mais ils doivent être revu et ne peuvent être utilisé par la suite.

11 Conclusions

11.1 Conclusion du projet

Concernant l'analyse des moyens de réalité augmentée dans le milieu sportif nous avons pu constater que beaucoup de chose se faisait sans que le spectateur s'en aperçoive car elle est intégrée de manière subtile et habile. Mais lorsque nous nous concentrons uniquement sur le cyclisme, nous constatons que ce sport est resté à l'écart dans l'utilisation de la réalité augmentée pour le spectateur. Des appareils sont en développement pour aider l'athlète à s'entrainer ou s'orienter mais rien ne se passe du côté de l'auditoire. Nous avons donc à ce niveau une innovation certaine à proposer.

Pour les outils de réalité augmentée, il en existe énormément mais comme l'état de l'art nous l'a montré seulement neuf ont retenu notre attention, et sur l'ensemble de ces outils sélectionnés uniquement quatre se sont distingués. Et encore après comparaison, deux se sont élevés au-dessus des autres en particulier, pour rappel il s'agissait de Kudan et de Vuforia. Deux raisons nous ont fait choisir l'outil Vuforia, la première est que l'entreprise Adventures-Lab l'emploie et la seconde est que nous possédions déjà une expérience d'utilisation de ce Framework.

Ensuite nous avons vu les moteurs de jeux et les logiciels de création d'applications. Unity a retenu notre attention car son utilisation est gratuite et dispose d'une très grande communauté. Tout comme Unity, Unreal Engine 4 est gratuit et la communauté est également conséquente mais l'aspect qui a fait pencher la balance est que la société Adventures-Lab utilise Unity.

11.2 Conclusion personnelle

Ce travail a été un défi en soit à mes yeux. Ayant déjà eu l'opportunité de travailler avec Unity ainsi qu'avec le Framework Vuforia je n'étais pas lâché en territoire totalement inconnu. Mon plus gros challenge a été de trouver une façon de transposer les coordonnées GPS au monde de Unity. Après quelques recherches j'ai finalement tranché sur le fait que la normalisation des données était l'une des meilleures solutions afin d'aboutir à des résultats positifs. L'extraction de la carte en relief depuis Google Earth m'a demandé quelques recherches sur les outils appropriés car bon nombre de solutions n'étaient pas en adéquation avec le résultat voulu. Le travail était de fournir une application en réalité augmentée qui affichait sur une carte, représentant la course, la position des cyclistes. Le prototype répond à ces critères et est fonctionnel. Il faudra néanmoins apporter quelques modifications visuelles comme spécifié sous la section Améliorations du prototype.

Ce projet m'a permis d'approfondir mes connaissances dans Unity et la réalité augmentée. Mais également d'appliquer des techniques Agile afin de gérer ce travail. En sachant que ce prototype sera certainement repris par la suite, il a également fallu écrire le code afin qu'il soit simple pour un tiers de reprendre le développement sans devoir tout recommencer de zéro.

12 Références et sources

Android Studio. (s.d.). *Android Studio features*. Récupéré sur developer.android.com: https://developer.android.com/studio/features.html

Appcelerator. (s.d.). *Appcelerator*. Récupéré sur www.appcelerator.com: https://www.appcelerator.com/

Apple. (s.d.). *Xcode*. Récupéré sur https://developer.apple.com/: https://developer.apple.com/xcode/

ARLab. (s.d.). *ARLab*. Récupéré sur http://www.arlab.com/: http://www.arlab.com/

ARToolKIt. (s.d.). *ARToolKit*. Récupéré sur artoolkit.org: https://artoolkit.org/documentation/doku.php?id=3_Marker_Training:marker_multi

Augment. (2017, 05 24). *Augmented reality developers, fear not, there is an alternative to Metaio*. Récupéré sur http://www.augment.com: http://www.augment.com/blog/augmented-reality-developers-fear-not-alternative-metaio/

Augmented Media. (2011, 09 18). *Une brève histoire de la réalité augmentée*. Récupéré sur augmentedmedia.net: https://augmentedmedia.net/2011/09/18/une-breve-histoire-de-la-realite-augmentee/

azoft. (2016, 03 01). *azoft*. Récupéré sur http://cases.azoft.com: http://cases.azoft.com/top-5-tools-creating-augmented-reality-apps/

commentcamarche. (2017, 05 17). *Réalité augmentée : avantages et exemples d'utilisation*. Récupéré sur http://www.commentcamarche.net: http://www.commentcamarche.net/faq/46569-realite-augmentee-avantages-et-exemples-d-utilisation

developereconomics. (2015, 03 16). *Top 5 Tools for Augmented Reality in Mobile Apps*. Récupéré sur www.developereconomics.com: https://www.developereconomics.com/top-5-tools-for-augmented-reality-in-mobile-apps

Dico du web. (s.d.). *Service web - Web Service*. Récupéré sur http://www.dicodunet.com: http://www.dicodunet.com/definitions/normes/service-web.htm

DroidAR. (s.d.). *DroidAR*. Récupéré sur https://bitstars.github.io: https://bitstars.github.io/droidar/

français, D. (s.d.). *Dictionnaire Français*. Récupéré sur http://www.linternaute.com: http://www.linternaute.com/dictionnaire/fr/definition/gps/

Google Glass Apps. (s.d.). *Google Glass Application List*. Récupéré sur http://glass-apps.org: http://glass-apps.org/google-glass-application-list

GPST. (s.d.). *GEO POSITIONING SWISS TECHNOLOGY*. Récupéré sur http://www.gpst.ch/: http://www.gpst.ch/

homecine-compare. (s.d.). *Définition Kinect*. Récupéré sur www.homecine-compare.com: http://www.homecine-compare.com/definition-de-kinect.htm

huffingtonpost. (2016, 17 08). *huffingtonpost*. Récupéré sur http://www.huffingtonpost.com: http://www.huffingtonpost.com/entry/did-sports-really-pave-the-way-for-augmented-reality_us_57b4889be4b03dd53808f61d

JDN. (2017, 07). *Framework, définition, traduction*. Récupéré sur http://www.journaldunet.com/: http://www.journaldunet.com/solutions/pratique/dictionnaire-du-webmastering/outils/19466/framework-definition-traduction.html

JetBrains. (s.d.). *AppCode*. Récupéré sur https://www.jetbrains.com: https://www.jetbrains.com/objc/

Kudan. (s.d.). *An Introduction to Simultaneous Localisation and Mapping - See more at: https://www.kudan.eu/kudan-news/an-introduction-to-slam/#sthash.OI870zCW.dpuf*. Récupéré sur www.kudan.eu: https://www.kudan.eu/kudan-news/an-introduction-to-slam/

Kudan. (s.d.). *Kudan*. Récupéré sur www.kudan.eu: https://www.kudan.eu/

LayAR. (s.d.). *layar*. Récupéré sur https://www.layar.com/: https://www.layar.com/

metaio. (s.d.). *Product Support*. Récupéré sur http://www.metaio.eu: http://www.metaio.eu/product_support.html

Microsoft. (s.d.). *Developpeur Microsoft*. Récupéré sur https://developer.microsoft.com: https://developer.microsoft.com/fr-fr/windows/apps/getstarted

Microsoft. (s.d.). *Téléchargements et outils pour Windows 10*. Récupéré sur https://developer.microsoft.com: https://developer.microsoft.com/fr-fr/windows/downloads

Microsoft. (s.d.). *Universal Windows Platform documentation*. Récupéré sur https://docs.microsoft.com: https://docs.microsoft.com/en-us/windows/uwp/index

NIST. (2014, 02 13). *C'est quoi le CLOUD*. Récupéré sur Culture Informatique: http://www.culture-informatique.net/cest-quoi-le-cloud/

OpenClassrooms. (2017, 06 20). *Utilisez des API REST dans vos projets web*. Récupéré sur https://openclassrooms.com/: https://openclassrooms.com/courses/utilisez-des-api-rest-dans-vos-projets-web/qu-est-ce-qu-une-api

Realite-Virtuelle. (2014, 03 31). *L'histoire de la réalité virtuelle*. Récupéré sur http://www.realite-virtuelle.com: http://www.realite-virtuelle.com/lhistoire-realite-virtuelle

rslnmag. (2016, 05 2). *Réalité mixte, virtuelle ou augmentée : de quoi parle-t-on ?* Récupéré sur rslnmag.fr: https://rslnmag.fr/innovation/realite-mixte-virtuelle-augmentee-definition/

SocialCompare. (2017, 07 04). *Augmented Reality SDK Comparison*. Récupéré sur http://socialcompare.com: http://socialcompare.com/fr/comparison/augmented-reality-sdks

Softpedia. (2015, 08 05). *metaio SDK*. Récupéré sur Softpedia: http://www.softpedia.com/get/Programming/SDK-DDK/metaio-Mobile-SDK.shtml

Solos. (s.d.). *Solos Smart Cycling Glasses with Heads Up Micro-Display*. Récupéré sur https://www.kickstarter.com: https://www.kickstarter.com/projects/1101608300/solos-smart-cycling-glasses-with-heads-up-micro-di

SportsWearable. (2016, 12 08). *sportswearable*. Récupéré sur http://www.sportswearable.net: http://www.sportswearable.net/raptor-ar-smartglasses-data-provider-cyclists/

techcrunch. (2015, 05 28). *Apple Acquires Augmented Reality Company Metaio*. Récupéré sur https://techcrunch.com/: https://techcrunch.com/2015/05/28/apple-metaio/

The Telegraph. (2010, 06 15). *Wimbledon 2010: IBM launches augmented reality app for tennis fans*. Récupéré sur http://www.telegraph.co.uk: http://www.telegraph.co.uk/technology/mobile-phones/7829722/Wimbledon-2010-IBM-launches-augmented-reality-app-for-tennis-fans.html

Unity. (s.d.). *Unity*. Récupéré sur https://unity3d.com/fr: https://unity3d.com/fr

Unreal Engine. (s.d.). *Unreal Engine*. Récupéré sur https://www.unrealengine.com: https://www.unrealengine.com/what-is-unreal-engine-4

UNREAL4AR. (s.d.). *Unreal4AR*. Récupéré sur http://www.unreal4ar.com/: http://www.unreal4ar.com/

upsidelearning. (2010, 04 30). *upsidelearning*. Récupéré sur www.upsidelearning.com: https://www.upsidelearning.com/blog/index.php/2010/04/30/tools-for-developing-augmented-reality-applications/

Vox. (2016, 02 06). *youtube*. Récupéré sur www.youtube.com: https://www.youtube.com/watch?v=1Oqm6eO6deU

Vuforia. (s.d.). *Vuforia*. Récupéré sur www.vuforia.com: https://vuforia.com

Vuforia. (s.d.). *Vuforia Developer Portal*. Récupéré sur www.developer.vuforia.com: https://developer.vuforia.com/forum/unity-extension-technical-discussion/vuforia-6-uwp

Wikipedia. (s.d.). *Framework*. Récupéré sur fr.wikipedia.org: https://fr.wikipedia.org/wiki/Framework

Wikipedia. (s.d.). *Kinect*. Récupéré sur Wikipedia: https://fr.wikipedia.org/wiki/Kinect

Wikipedia. (s.d.). *Liste de moteurs de jeu*. Récupéré sur www.wikipedia.com: https://fr.wikipedia.org/wiki/Liste_de_moteurs_de_jeu

Wikitude. (s.d.). *Wikitude*. Récupéré sur http://www.wikitude.com: http://www.wikitude.com/products/wikitude-sdk/

Annexes

Annexes I Product Backlog de la recherche et du prototype

1,2,3,5,8,1
3,21

US Nr.	Theme	En tant que	User Stories J'aimerai pouvoir	afin que/de	Priorité	Story Points	Critère d'acceptance	Statut	Remarques
1	Prototype	Développeur/Responsable de projet	créer une classe objet de type participant	d'avoir toutes les informations nécessaire à chaque participant dans un objet	1000	5			
2	Prototype	Développeur/Responsable de projet	instantier mes objets participant avec les valeurs du csv importé	utiliser les données de ces objets	975	3			
3	Prototype	Développeur/Responsable de projet	créer de manières dynamique un GameObject qui représente un participant	pouvoir l'afficher sur le monde	950	2			
4	Prototype	Développeur/Responsable de projet	trouver une formule qui transpose des coordonnées gps en coordonnées x,y,z	afin de pouvoir appliquer cette formule sur les données gps des participants	925	21			
5	Prototype	Développeur/Responsable de projet	récupérer un fichier csv depuis l'ordinateur local	pouvoir assigner les variables de mon objet participant	900	8			
7	Prototype	Développeur/Responsable de projet	récupérer depuis un webservice un fichier csv	pouvoir assigner les variables de mon objet participant	875	13			
8	Prototype	Développeur/Responsable de projet	implémenter la carte en 3D du circuit de la course	avoir un visuel concret sur mon projet	850	2			
9	Prototype	Développeur/Responsable de projet	définir un visuel pour les participants	avoir une meilleur vu sur l'implémentation de la position	825	3			
10	Prototype	Développeur/Responsable de projet	interactions entre l'utilisateur et les informations des participants	de rentre l'expérience utilisateur plus concrète	800	13			interaction a définir en fonction de l'affichage et de

1,2,3,5,8,13,21

U Nr.	Theme	User Stories En tant que	J'aimerai pouvoir	afin que/de	Priorité	Story Points	Critère d'acceptance	Statut	Remarques
1	Recherche	Responsable de projet	analyser les moyens utiliser dans le domaine du sport et plus précisément les courses de cyclisme	connaitre l'était de ce qu'il se fait actuellement	1100	8			
2	Recherche	Responsable de projet	lister une série d'outils de AR*	avoir une vue d'ensemble des outils à disposition	1000	13	Liste des moyens de AR		*réalité augmentée
3	Recherche	Responsable de projet	définir chaque outils	pouvoir les comparer	970	5	Liste des moyens de AR les plus récurrents		
4	Recherche	Responsable de projet	établir un comparatif des outils de AR	d'éliminer les outils les moins intéressants et de pouvoir faire un choix sur l'outil le plus adapté	960	8	Comparatif complet		
5	Recherche	Responsable de projet	lister une série de logiciels	connaître les environnements de développement possible	950	8	Définir les plateformes de développement/moteur de jeux		
6	Recherche	Responsable de projet	définir le concept de AR	de pouvoir établir un comparatif	900	2	lecteur a compris ce qu'est la AR		
7	Recherche	Responsable de projet	définir le concept de VR*	de pouvoir établir un comparatif	850	2	lecteur a compris ce qu'est la VR		*réalité virtuelle
8	Recherche	Responsable de projet	éclaircir le concept de réalité mixte	d'avoir une meilleur approche de ce concept	800	2	lecteur a compris le concept de réalité mixte		
10	Recherche	Responsable de projet	Proposé la meilleur manière d'afficher les athlètes sur une carte en 3D	d'avoir une approche concrète et réaliste d'un prototype	700	13	la solution d'affichage présenté est accepté par le client		

Déclaration de l'auteur

Je déclare, par ce document, que j'ai effectué le travail de Bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de Bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après : M. Pierre-Albert Cantin.